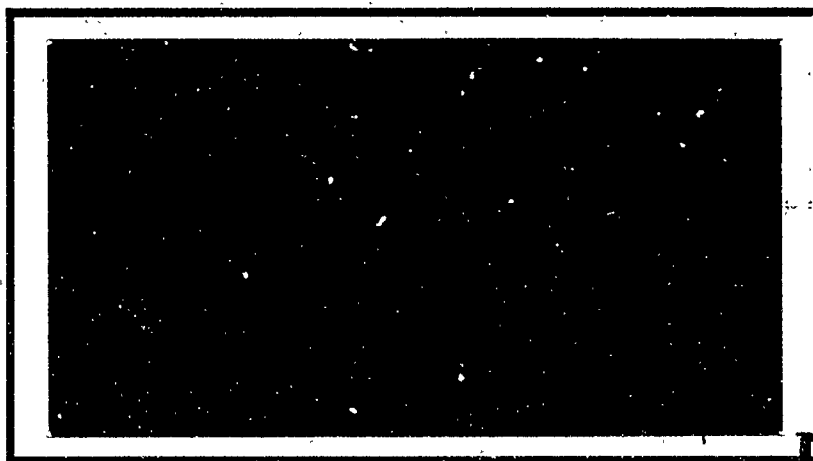


DTIC FILE COPY

AD-A230 386



DTIC
ELECTE
JAN 08 1991
S E D

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

91 1 3 137

AFIT/GE/ENG/90D-62

ANGLE OF ARRIVAL DETECTION
THROUGH
ARTIFICIAL NEURAL NETWORK
ANALYSIS OF
OPTICAL FIBER INTENSITY PATTERNS

THESIS

Scott Thomas
Captain, USAF

AFIT/GE/ENG/90D-62

DTIC
ELECTE
JAN 08 1991
S E D

Approved for public release; distribution unlimited.

AFIT/GE/ENG/90D-62

ANGLE OF ARRIVAL DETECTION
THROUGH
ARTIFICIAL NEURAL NETWORK
ANALYSIS OF
OPTICAL FIBER INTENSITY PATTERNS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Scott Thomas, B.S.E.E
Captain, USAF

December, 1990

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited.



Preface

This thesis demonstrated that a detection system using the output intensity pattern of a short piece of optical fiber could be used to classify angles of arrival of incident laser energy to one-tenth of a degree.

This research effort would not have been possible if it was not for the tremendous amount of help I received while attending the Air Force Institute of Technology. First off, I would like to thank my professors and fellow students who without their help I would never have gotten to the point of writing this thesis. Professors such as Lt Col Meer, Maj Tatman, Capt Welsh, and Marty DeSemio all who put the extra time and effort in to help me through my coarse work. I am also indebted to my fellow students Becky Seeger, Joe Brickey, Mark Brown, Van Osborne, Rick Ricart, Vicki Sundberg, Al L'Homme, Vance McMillan, Scott Stowe, and Daniel Zahirniak.

I'm especially grateful to my thesis advisor, Dr. Steven Rogers, who's devotion to knowledge allowed me to bridge the gap between ideas and research and to my thesis committee members Dr Matthew Kabrisky and Lt Col Norman.

I also want to thank my predecessor Captain John Welker for his work into angle of arrival detection and for his obtaining the necessary equipment so that I could carry the research effort to the next level.

I would like to thank my parents for giving me the ability to succeed in this world and most of all my wife Vicky and my children Scott Jr, Angela, David, and Jennifer for giving me the love and caring I've needed to keep going.

Scott Thomas

Table of Contents

	Page
Preface	ii
Table of Contents	iii
List of Figures	vi
List of Tables	vii
Abstract	viii
 I. Introduction	 1
1.1 Background	1
1.2 Problem Statement	3
1.3 Definitions	3
1.4 Assumptions	4
1.5 Scope	4
1.6 Approach	5
1.7 Organization	7
1.8 Summary	9
 II. Summary of Current Knowledge	 10
2.1 Non-Imaging Systems	12
2.2 The Angle of Arrival Meter	12
2.3 The Common Opto-Electronic Laser Detection Systems . .	13
2.4 The Optical Sensor with High Directional Resolution . . .	15
2.5 AOA Detection through Analysis of Optical Fiber Intensity Patterns	17
2.6 Summary	18

	Page
III. Methodology	19
3.1 Optical Fiber	19
3.2 Data Reduction	20
3.3 Laboratory Setup	23
3.4 Phase I	25
3.5 Simplifying the Methodology	26
3.6 Data Collection	27
3.7 Phase II	30
3.8 Radial Basis Function	32
3.8.1 Matrix Inversion	34
3.9 Software	36
3.10 Summary	38
IV. Findings	39
4.1 Baseline	39
4.2 Rotational Stage	40
4.3 Processing Simplification	40
4.4 Quarter Degree Results	46
4.5 Fifth of a Degree Results	46
4.6 Phase II Results	49
4.6.1 RBF and 0.1deg AOA Increments	51
4.7 Normalized Data	52
4.8 One Tenth of a Degree AOA	54
4.9 Summary	55
V. Conclusions and Recommendations	56
5.1 Baseline Conclusions	56
5.2 Phase I Conclusions	56

	Page
5.3 Phase II Conclusions	57
5.4 Recommendations for Further Research	58
Appendix A. Alignment Procedure	61
A.1 Alignment Tool	61
A.2 Laser Beam Alignment	63
A.3 Attenuator Adjustment	63
A.4 Zero Degree Baseline	63
Appendix B. Collimator	65
Appendix C. Spiricon File	67
Appendix D. Program XFM	69
D.1 Running the Program	70
D.2 Example Run	71
Appendix E. Subroutine FOURN	78
Appendix F. Program NACDIS	83
F.1 Running the Program	83
F.2 Example Run	84
Appendix G. Program MZNET	91
G.1 Running the Program	91
Bibliography	94
Vita	96

List of Figures

Figure	Page
1. Structure of an Insect Eye	2
2. AOA Detection System	6
3. Simplified Spectrometer System	12
4. Simplified Angle of Arrival Meter System	13
5. Simplified COLD System	14
6. Simplified Staring Optical Sensor System	16
7. Simplified Intensity AOA Detection System	17
8. Ray Optics	20
9. Intensity verse Angle of Arrival	21
10. Before Reduction	22
11. After Reduction	23
12. Laboratory Setup	24
13. Error Associated with Feature Vectors	30
14. Two Class Example	31
15. Radial Basis Function Network	33
16. Example of a Two-dimensional RBF	34
17. Magnitude Plot of a Typical Fourier Transform	44
18. Alignment Tool	62
19. Zero Degree Intensity Pattern	64
20. Zero Degree Intensity Slice	64
21. Collimator	66
22. Spiricon File	68
23. Input Array	78
24. Output Array	79

List of Tables

Table	Page
1. Baseline Results	40
2. Comparison of Processing Methods	42
3. Results 49 Component Feature Vector	43
4. Results 25 Component Feature Vector	45
5. Results 24 Component Feature Vector	45
6. Quarter Degree Results	47
7. Series 30 (0.2deg) Results	48
8. Series 50 (0.2deg) Results	48
9. Series 90 (0.2deg) Results	49
10. RBF 0.2deg Data Results	51
11. RBF 0.1deg Data Results	51
12. Accuracy vs Varying # of Training Exemplars	52
13. Normalized Series 30 (0.2deg) Results	52
14. Normalized Series 50 (0.2deg) Results	53
15. Normalized Series 90 (0.2deg) Results	53
16. Non-Normalized 0.1deg AOA Results	54
17. Normalized 0.1deg AOA Results	54

Abstract

This thesis demonstrated that an intensity pattern out of a short piece of optical fiber could be used to determine the angle of arrival (AOA), to within 0.1deg, of the incident laser energy on the front of the optical fiber. The optical fiber was a one-inch-long, 3mm-diameter, multimode, step-index, plastic fiber. The optical fiber was mounted to the front end of a charge injection device (CID) camera. The CID camera's angle with respect to the incident laser energy, a uniform amplitude plan wave, could be varied by a computer controlled rotational stage. The output of the CID camera was captured by Spiricon software. Captured outputs representing various AOAs were processed to provide template or test feature vectors. The processing method used a fast Fourier transform routine to create a 24 component low frequency feature vector. Two classification methodologies were used: a Euclidean distance method and a radial basis function (RBF) neural network.

(KR)

↑

ANGLE OF ARRIVAL DETECTION THROUGH ARTIFICIAL NEURAL NETWORK ANALYSIS OF OPTICAL FIBER INTENSITY PATTERNS

I. Introduction

The optical sensors of United States Air Force reconnaissance vehicles, such as satellites, are subject to temporary or permanent blinding from hostile (or threat) laser radiation. By detecting and determining the angle of arrival (AOA) of the hostile radiation, the reconnaissance vehicle may be able to protect its optical sensors by taking evasive maneuvers or by shutting down the optical sensors (such as closing a shutter) until the threat has passed. In addition, the vehicle can relay information to its ground terminal allowing the intelligence community to determine the source of the hostile laser radiation (15:1).

1.1 Background

The farther away a reconnaissance vehicle is from a threat source the harder it is for the vehicle to determine the relative angle of the threat laser radiation since the laser radiation spreads out. Therefore, to provide any useful AOA information, the reconnaissance vehicle must be able to determine the angle of arrival of the incident laser energy to less than one degree (3:7).

Accuracy is not the only requirement of AOA detectors used on reconnaissance vehicles. Reconnaissance vehicles, such as satellites, operate in the harsh environment of space where maintaining the stability of mechanical detection systems is

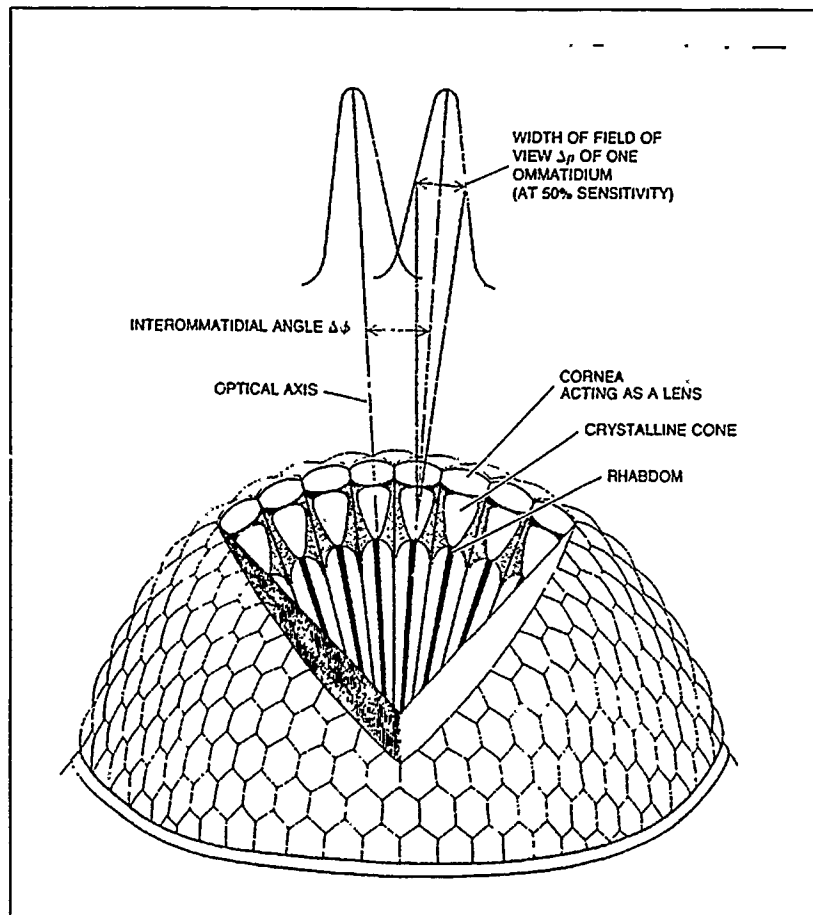


Figure 1. Structure of an Insect Eye (2:111)

difficult. Thus, detection systems for use on satellites should avoid mechanical sensors (3:i).

One such nonmechanical laser detection system is based on the structure of the compound eye of an insect (3:i). A compound eye is made up of many individual photoreceptors called ommatidia, in which each ommatidium points in a different direction. The ommatidia of the compound eye are spread out over a hemispheric dome, giving the insect a 180 degree field of view (FOV) around each eye. Each ommatidium focuses the light down to a common light-sensitive organ called the rhabdom (2:108). The structure of the compound eye of an insect is shown in Figure 1.

Two systems using the compound eye type of AOA sensor are the Staring

Optical Sensor and the Common Opto-Electronic Laser Detection System (both of these systems are explained in detail in Chapter II). Although both of these systems use different types of individual photoreceptors and different processing methods to determine the AOA of incident laser energy, they both use photoreceptors spread across a hemispheric dome to provide the required FOV.

The above systems demonstrate that a system built on the compound eye concept can be used to determine the AOA. This thesis will concentrate on one particular method of processing the information taken from one type of photoreceptor. The method used in this thesis is the AOA detection through the analysis of optical fiber intensity patterns. The photoreceptor used in this thesis is a one-inch-long, 3mm-diameter optical fiber.

The method of AOA detection through the analysis of optical fiber intensity patterns was first proposed by Capt Cole in his patent application for an AOA detector (15:5). Capt Welker, using Capt Cole's proposal, showed that the AOA (on a single one-inch-long, 3mm-diameter optical fiber) could be determined to within one degree by using pattern recognition techniques. Capt Welker's method is explained in more detail in Chapter II.

1.2 Problem Statement

To react and minimize optical damage to United States Air Force reconnaissance vehicles from a laser attack, the angle of arrival of the laser's energy must be determined to less than one degree. This research will determine how well an artificial neural network can determine the angle of arrival of incident laser energy from the analysis of the intensity pattern out of an optical fiber photoreceptor.

1.3 Definitions

Feature Vector A group of numbers characterizing a certain AOA. For this thesis effort, the feature vector will be a portion of the Fourier transformed intensity

pattern out of the optical fiber.

Template Vector A feature vector which represents a known AOA intensity pattern out of the optical fiber. Template vectors form the basis set of feature vectors.

Test Vector A feature vector used to test the accuracy of the AOA determination method.

Artificial Neural Network The grouping of simple computational elements (for this thesis radial basis functions) into a network that performs parallel processing to solve a problem (7:4). For this thesis effort, the problem is determining the AOA of a given laser threat.

1.4 Assumptions

Two assumptions are made in this thesis: 1) the one inch optical fiber provides the best intensity patterns to work from and 2) the results of doing AOA determination on a single fiber could be expanded to a compound eye made up of many fibers. The first assumption is based on Capt Welker's work that showed the one inch fiber provided a worst case accuracy of one degree whereas a two inch fiber provided a worst case accuracy of seven degrees. The second assumption is based on the fact that the Staring Optical Sensor System and the Common Opto-Electronic Laser Detection System both used similar individual photoreceptors for their optical sensors. Assumption two allows this research to be extended to a system having a FOV of 180 degrees.

1.5 Scope

As explained in Section 1.1, the structure of the compound eye has two main parts: the ommatidium (the individual photoreceptor) and the rhabdom (the common light sensitive organ). Many theories exist on what is the best way to simulate the parts of the compound eye. The theories on simulating the ommatidium range

from trying to copy the structure of the ommatidium with a small lens and an optical fiber to just using a hollow tube with a highly polished interior. Theories on simulating the rhabdom vary with the type of incident laser energy the system is trying to detect. This research will not determine what is the best way to simulate the compound eye; but, rather this research is going to concentrate on what is the best way to handle the data gained from one type of photoreceptor (the one-inch-long, 3mm-diameter optical fiber) and one type of light sensitive device (the charge injection device (CID) camera).

The main questions this thesis will address are; can improvements be made to Capt Welker's AOA detection method and how well can a radial basis function (RBF) neural network classify feature vectors. These feature vectors will be representing the various angles of arrival of incident laser energy on an optical fiber. Multiple template vectors, representing only a small portion of the FOV of the optical fiber photoreceptor, will be generated and used to train the RBF neural network. The RBF network will be setup to classify the separate training AOAs. Then ten randomly selected AOAs will be used to generate ten test vectors. These test vectors will be used to determine the accuracy of the RBF neural network at classifying previously unseen feature vectors. The classification accuracy of the neural network will be compared to the one degree accuracy baseline set by Capt Welker's work.

1.6 Approach

This section summarizes the sequence of events in this research. This research will be done in two phases. Phase I involves automating and reproducing Capt Welker's research to provide a baseline for phase II of this thesis. Phase II involves taking the data gathered in phase I and applying it to a radial basis function (RBF) artificial neural network to investigate how well the RBF neural network can classify the AOA data.

The first step of phase I will be to gather images produced by various angles

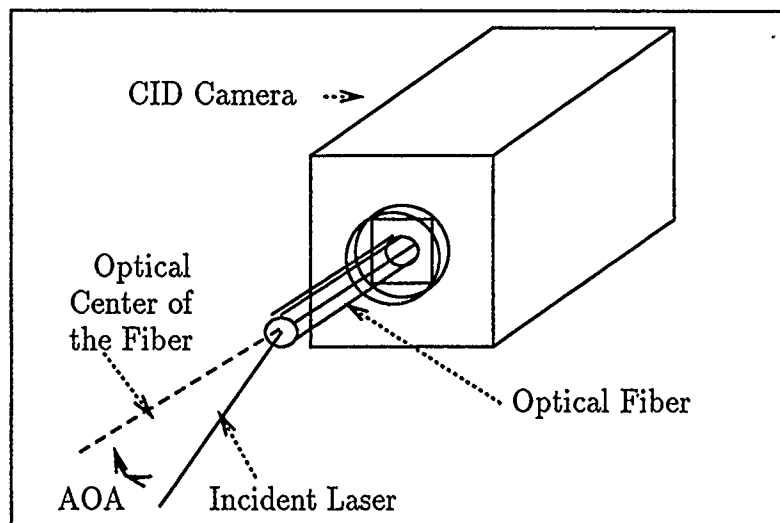


Figure 2. AOA Detection System

of arrival. The images will be obtained by moving the optical fiber (attached to the front of the CID camera, reference Figure 2) in relation to the fixed laser beam and capturing the images with the Spiricon 2250 Laser Beam Diagnostic System. The CID camera will be moved to each of the required AOAs with an automated rotational stage. The automated stage shall have an accuracy of 0.001 degree and will be stepped to any angle setting by a personal computer (PC) interface.

Each AOA image plus overhead Spiricon file information will be stored as a Spiricon picture file by the Spiricon software. Refer to Appendix C for the complete makeup of the Spiricon file. The center of each Spiricon file represents the image out of the CID camera's 512-by-512 pixel array. The center of the Spiricon file will be extracted and processed into a feature vector representing the image's AOA.

Three different versions of the AOA image will be compared to determine which version provides the best feature vector for classifying AOAs. The three versions to be compared are: 1) the entire 512-by-512 square array, 2) just the center 256-by-256

of the entire 512-by-512 square array, and 3) a reduced image using Capt Welker's PREPROC program.

The feature vector out of the Fourier transform routine will be a 49 component vector made up of the magnitude of the dc and lower three harmonic components of the Fourier transform of the intensity image out of the optical fiber. Different portions of the 49 component feature vector will be compared to determine which components provide the best feature vector for AOA classification.

Previous work by Capt Welker did most of the processing of the Spiricon files on the PC, which required transferring data to the Micro VAX and then back to the PC. This research will simplify the processing of the Spiricon files down into feature vectors by only using the PC to grab the intensity images out of the fiber and then perform all processing of the images on the Micro VAX III. Using the VAX to perform all processing will decrease data transfer time and speed data processing time.

Various FOVs of the optical fiber will be studied to determine how accurate the Euclidean distance algorithm is at classifying AOAs. Initially the approximate 30 degree FOV of the optical fiber will be studied and then the FOV will be reduced so as to allow more data sets to be repeated. Multiple data sets will show how reproducible the AOA classification methodology is. Multiple data sets will also be required for the training and testing of the RBF neural network of phase II.

Phase II will take the feature vectors of phase I and apply them to a three layer RBF neural network. The multiple run files, of a series, will allow many feature vectors to train the RBF neural network. The remaining feature vectors, of a series, will be used to test the RBF neural network's ability to classify previously unseen feature vectors. The network's accuracy will then be compared to the results of phase I.

1.7 Organization

This thesis is organized as follows:

- Chapter II presents a literature review of current research in angle of arrival of incident laser energy on an object.
- Chapter III covers the methodology used to gather the test data.
- Chapter IV presents the analysis for the test runs in the data gathering phase of this thesis effort.
- Chapter V presents the conclusions drawn from the analysis of the data and presents recommendations for further research.
- Appendix A provides information on how to setup and align the optic's of the AOA detection system.
- Appendix B provides information on the collimator use in the AOA detection system.
- Appendix C provides break down of the Spiricon picture file.
- Appendix D provides the source code for the program XFM. The XFM program was the final version of the program written to process the Spiricon files down into 49 component feature vectors.
- Appendix E provides the source code for the subroutine FOURN. The subroutine FOURN is used by the XFM program to Fourier transform a two-dimensional image array.
- Appendix F provides the source code for the program NACDIS. The program NACDIS was the final version of the program written to compute the Euclidean distance between various template and test feature vectors.
- Appendix G provides the source code for the program MZNET. The MZNET program converted the output of the XFM program into a format readable by the program used in phase II of this research.

1.8 Summary

An AOA detection system is required to protect USAF reconnaissance vehicles from damage due to threat laser radiation. This thesis effort will concentrate on ways to improve an AOA detection system using the intensity patterns out of a short piece of optical fiber. An unique intensity pattern is generated for each AOA of the laser radiation on the front of the optical fiber. This research will first try to answer how small of an AOA is detectable by using the Euclidean distance methodology to classify AOAs. Then will see if the classification can be improved by using the RBF neural network to classify AOAs.

II. Summary of Current Knowledge

Most of the research into angle of arrival (AOA) of incident laser radiation has been generated out of the military's need to detect, counter, and evade laser radars, laser fire control systems, and target illuminating lasers. Angle of arrival detectors fall within one of two main classes: imaging and non-imaging systems. The imaging methodology is simplest and gives a direct method of determining the AOA. The non-imaging methodology provides for a rugged simple front-end receiver, but requires a more elaborate processing scheme to determine the AOA. The non-imaging processing schemes determine the AOA by analyzing the intensity profile from an array of detectors.

The selected method, whether it is imaging or non-imaging, is "usually the result of maximizing one or more design parameters, such as sensitivity, the field-of-view or angle accuracy" (3:1). Sensitivity relates to the dynamic range of the system. For example, can the system detect low levels of laser radiation while not being blinded by higher levels? The field of view (FOV) specification of the system indicates how many degrees of azimuth or elevation the detector can monitor. Usually the accuracy of the monitor varies in azimuth verses elevation.

The above parameters are dictated by the type of threat laser radiation the AOA system is supposed to detect. For example, in the battle field environment the types of lasers encountered would be the most diverse; therefore, the hardest to design for. Where as in the scenario where the system is trying to prevent optical blinding, then sensitivity isn't an important parameter (since high levels of laser radiation would be required to harm the optics of an imaging system), but the field of view parameter would be important.

The FOV parameter is usually the parameter that increases the complexity of the system, since increasing the FOV increases the number of detectors in AOA

detection systems and increase the complexity of the AOA processing scheme. The FOV parameter is one that makes the imaging system stand out, since with one fisheye lens the system can monitor 180 degrees.

The basic imaging system measures the AOA directly from the difference between the focused image of the incident laser energy and the center of the focal plane of the image. Capt Robert D. Kaiser used a more sophisticated version of the above system, by combining the basic imaging properties of a fisheye lens and a spectrometer. Capt Kaiser states: "Fisheye lenses, unlike other lenses, have a linear relation between angle of arrival (AOA) and location of the image in the focal plane" (5:10). The AOA equals the distance from center of the focal plane divided by the focal length of the fisheye lens (5:10).

The combining of the imaging system with a spectrometer allowed not only the measuring of the AOA, but also the measuring of the incident laser's wavelength and energy. The basic system, refer to Figure 3, consisted of a fisheye lens (giving the system 180 degree field of view), imaging lenses, spectrometer, image detector, and detector controller (5:78). The imaging lenses focused the incident image/laser radiation onto the input of the spectrometer. The spectrometer breaks the incident image up into its spectra. The controller allowed for the freezing of a given image spectra for wavelength and AOA determination. The spectra of the different incident image sources will be separated in height on the output of the spectrometer. So, the laser's spectra will be separated in height on the output from the other incident energy in the scene, such as the sun's energy (5:5).

Capt Kaiser showed that the height of the spectra of the laser is directly related to its AOA; and that by using this method the AOA could be determined to one degree of accuracy

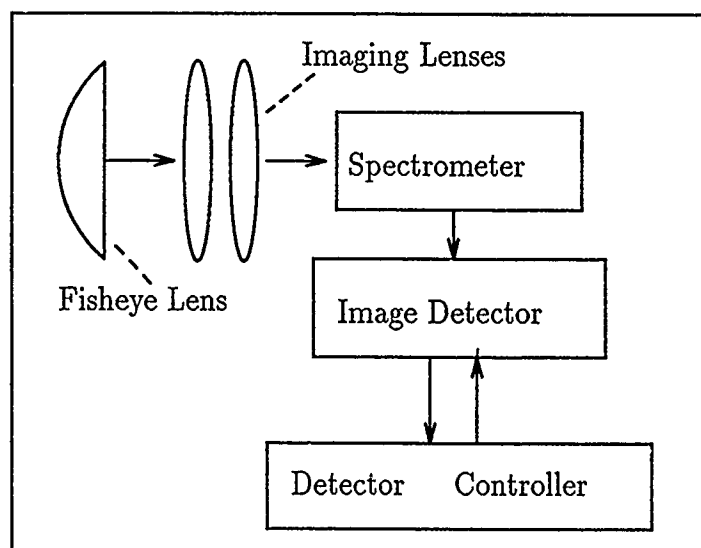


Figure 3. Simplified Spectrometer System (5:4)

2.1 Non-Imaging Systems

The research using non-image systems is by far the most extensive. The non-imaging system, although more complex electronically, offers the ruggedness required for military and space operations. Four such systems will be reviewed: 1) the angle of arrival meter by Herbert B. Holl, 2) the Common Opto-Electronic Laser Detection Systems (COLDS), 3) the optical sensor with high directional resolution, and 4) angle of arrival detection through analysis of optical fiber intensity patterns.

2.2 The Angle of Arrival Meter

The basic system invented by Herbert B. Holl consists of triangular reflector cube (with integral detector arrays) and an external AOA processor (1:1-2). As shown in Figure 4, the incident laser radiation enters the cube and then is reflected towards the detector arrays located on the three edges of the triangular reflector. The

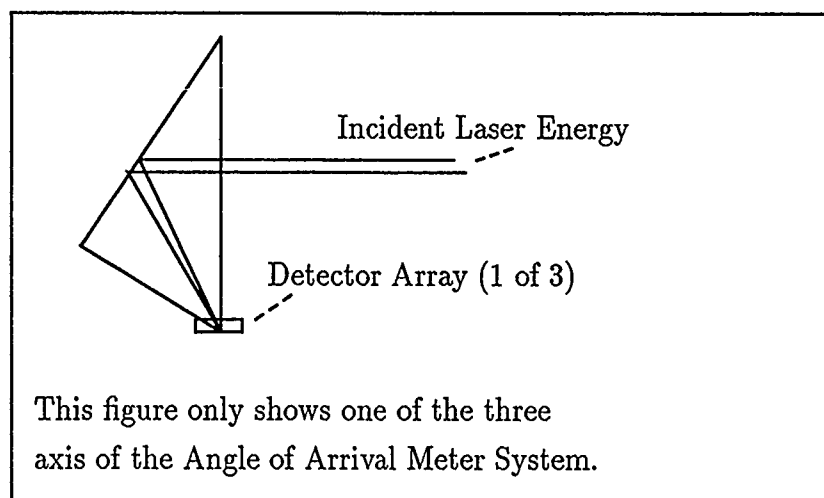


Figure 4. Simplified Angle of Arrival Meter System (1:14)

detectors send the measured incident energy onto the processor, which determines the horizontal and vertical components of the AOA, to an estimated fraction of a degree (1:10).

2.3 The Common Opto-Electronic Laser Detection Systems

The Common Opto-Electronic Laser Detection Systems (COLDS) system was developed by the Messerschitt-Bolkow-Blon (MBB) Corporation and was tested by the Air Force in the summer of 1985. The system could measure the pulse repetition frequency and the angle of arrival of incident laser energy (12:6).

The system, reference Figure 5, consisted of a hemispheric optical head, a fiber optic interconnect, an optical detector system, and an AOA determination algorithm. The design of the optical head provided an AOA detector system with a theoretical FOV of 180 degrees in azimuth and 190 degrees in elevation (12:6).

The optical head contain two different types of lens: four large FOV lenses

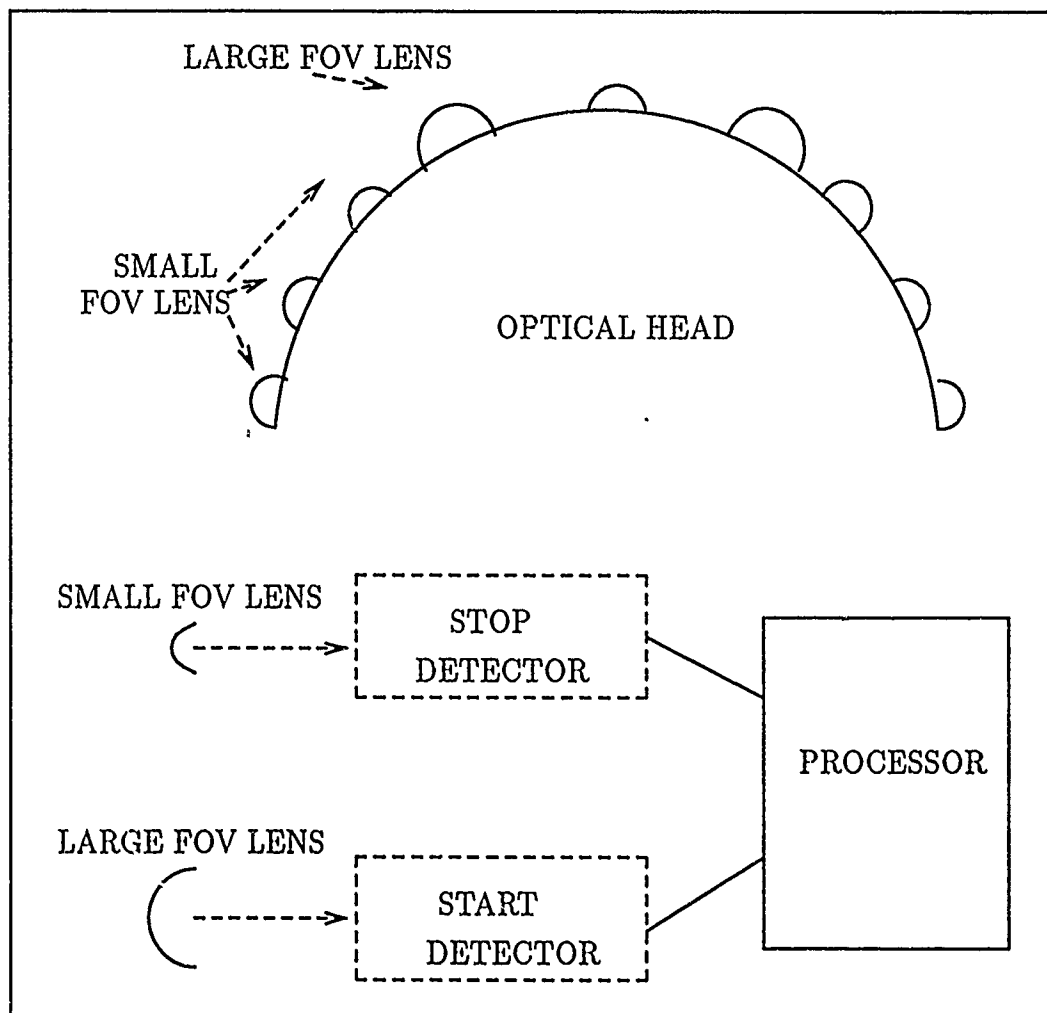


Figure 5. Simplified COLD System (12:3)

(one per quadrant) and many small FOV lenses (each small lens provided for a 22.5 degree overlapping FOV in azimuth). The light energy received by each lens was transferred to the optical detector system by fiber optic cables.

The optical detector system consisted of two different detectors, one for each type of lens. The fiber optic cable from the large FOV lenses terminated on the start detector, and the optical cable from the small FOV lenses terminated on the stop detectors. The start and stop signals from the detector system in conjunction with the AOA algorithm determined the AOA with a plus or minus five degree azimuth accuracy, the elevation accuracy was not tested (12:40).

2.4 The Optical Sensor with High Directional Resolution

The optical sensor with high directional resolution extends the concepts used by radar warning receivers to the problem of determining the angle of arrival of incident laser energy on an array of optical detectors. The system, the Staring Optical Sensor, was designed and built by the Interactive Intelligent Imagery Corporation. The following information is from the final report for phase 1 feasibility demonstration, dated march 1989. The basic system, reference Figure 6, consists of a hemispheric dome (with 225 apertures), a fiber optic interconnect, detector array, and an angle processor (3:i-ii). The hemispheric dome is fabricated in such a way as to allow each fiber of the fiber optic interconnect to view part of the 180 degree field of view of the dome (3:4). The fiber optic interconnect transfers the intensity of the incident laser energy on the dome to the detector array. The angle processor scans the detector array for the resulting intensity (amplitude) pattern and by using amplitude comparison techniques determines the AOA (3:33). Test results show that the system could determine the AOA "to better then one degree" (3:23).

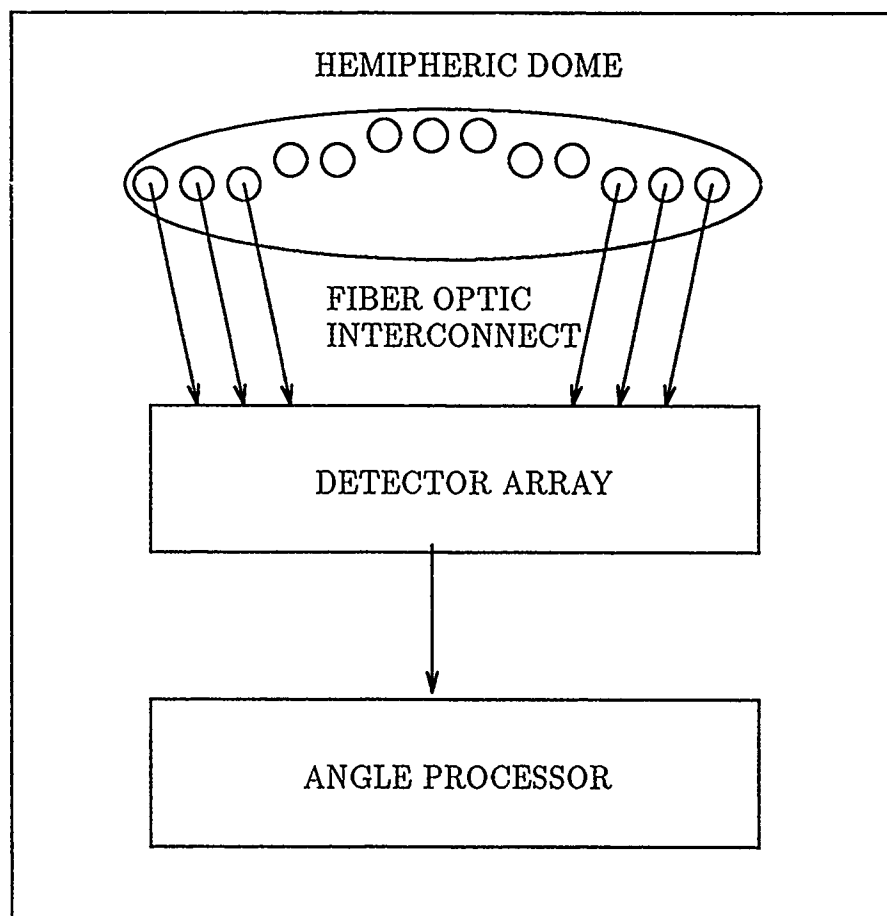


Figure 6. Simplified Staring Optical Sensor System (3:4)

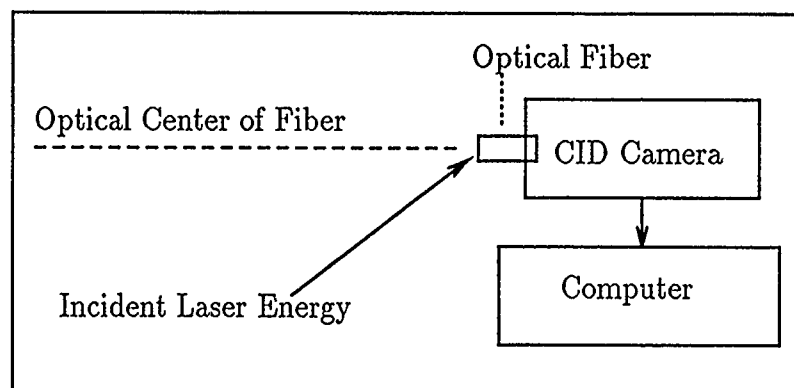


Figure 7. Simplified Intensity AOA Detection System (15:4)

2.5 AOA Detection through Analysis of Optical Fiber Intensity Patterns

This research was accomplished by Capt John Wallace Welker in 1989 as part of his master's degree program. Capt Welker showed that the angle of arrival could be determined within one degree by using pattern recognition techniques. The basic system, reference Figure 7, used consisted of a short piece of 3mm fiber optic cable (1, 2, and 3 inch lengths were tested), a CID camera, and a personal computer (PC) with frame grabber software.

One end of the optical fiber was illuminated by a laser beam at various angles to the optical center of the fiber. Due to the light transfer characteristics of the fiber used an unique intensity profile was produced on the camera end of the fiber, with each change in the AOA of the incident laser on the other end of the fiber. The intensity profile was captured by the CID camera in conjunction with the frame grabber software running on the PC.

The captured data was reduced, by using Fourier transfer techniques, down into a feature vector representing a given captured intensity pattern. The feature vector was then compared with the know AOA intensity patterns using an Euclidian

distance measurement between the given and known feature vectors. The comparison which resulted in the least difference indicated the AOA (15:1-17).

2.6 Summary

This chapter summarized five different angle of arrival detectors (one imaging and four non-imaging). The imaging system (by Capt Kaiser) provided not only the AOA of the incident laser radiation, but also its wavelength and energy level. The system easily provided AOA accuracy to one degree (azimuth), without intensive processing algorithms.

The four non-imaging AOA detection systems are diverse and all use different methods to determine the AOA. It should be noted that the Angle of Arrival Meter information is from a patent application. The last three AOA detector systems have been prototyped giving AOA accuracy figures from one degree to five degrees. Since the accuracy figures given in the researched data is from prototypes an in-depth comparison of the systems can not be made. Each system seems to be capable of highly accurate angle of arrival detection, if the lessons learned from the prototypes were incorporated into a production product.

III. Methodology

This research was based on pattern recognition techniques to determine the AOA of laser energy on a photoreceptor (one-inch-long, 3mm-diameter optical fiber). The idea is to "match some measurement to an internal description, template, of the quantities of interest." (4:4) The measurement in this research is a portion of the Fourier transform of the intensity pattern out of the optical fiber. The basic idea is to recognize a pattern or a group of patterns which uniquely classify each possible AOA. In the first phase of this research the methodology is to generate a template file containing one template for each AOA. Each unknown or test measurement is then compared to the template file and the closest match classifies the unknown AOA. In the second phase of this research the methodology is to combine multiple measurements of each AOA along with a radial Gaussian probability function to create a network capable of classifying unknown AOAs.

3.1 Optical fiber

Due to the light transfer characteristics of the one-inch-long, 3mm-diameter optical fiber an unique intensity pattern was generated on the camera end of the fiber for each AOA of the laser energy. This unique pattern is caused by meridional and to a greater extent skew rays traveling down the fiber. Meridional rays follow a zig-zag path along the fiber and always stay within a symmetrical plane containing the axis of the fiber. The skew rays do not travel down the fiber in just one plane but follow a helical type path through the fiber (6:23-24).

The optical fiber used in this research is a multimode step-index fiber. The multimode step-index fiber has an uniform refractive index core surrounded by a lower refractive index cladding. The refractive index of the core was $n_{core} = 1.492$. The refractive index of the cladding was $n_{cladding} = 1.416$.

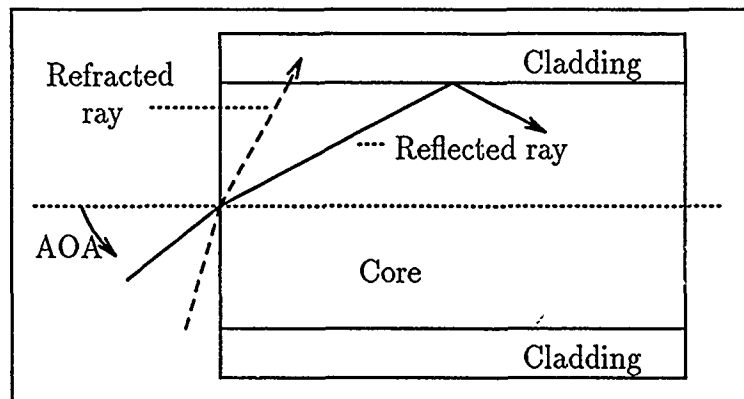


Figure 8. Ray Optics (6:24)

As light energy enters the fiber it bends towards the axis of the fiber due to the higher refractive index of the fiber versus the air. As the light energy propagates down the fiber, the lower index of refractive of the cladding causes the light energy propagating towards the cladding to bend towards the high refractive index of the core. At a certain angle of incidence the light energy no longer is completely reflected but some of the light energy is refracted into the cladding (6:23-25). Refer to Figure 8. Figure 9 shows how the intensity out of the fiber varies as the AOA. The graph shown was created by taking the dc components of the Fourier transform of the template AOAs from 0 to 29 degrees. At approximately 28 degrees the intensity patterns out of the fiber become too weak for the AOA detection methods used in this research to work effectively. This research also indicated that between 0 and 1 degrees the intensity pattern would change erratically. Therefore, based on the intensity level used in this research, the best angular range or FOV for a single fiber is approximate 1 through 26 degrees.

3.2 Data Reduction

In this research the quantities of interest are the values of the 512-by-512 unsigned bytes of picture information out of the Spiricon picture file (refer to Appendix

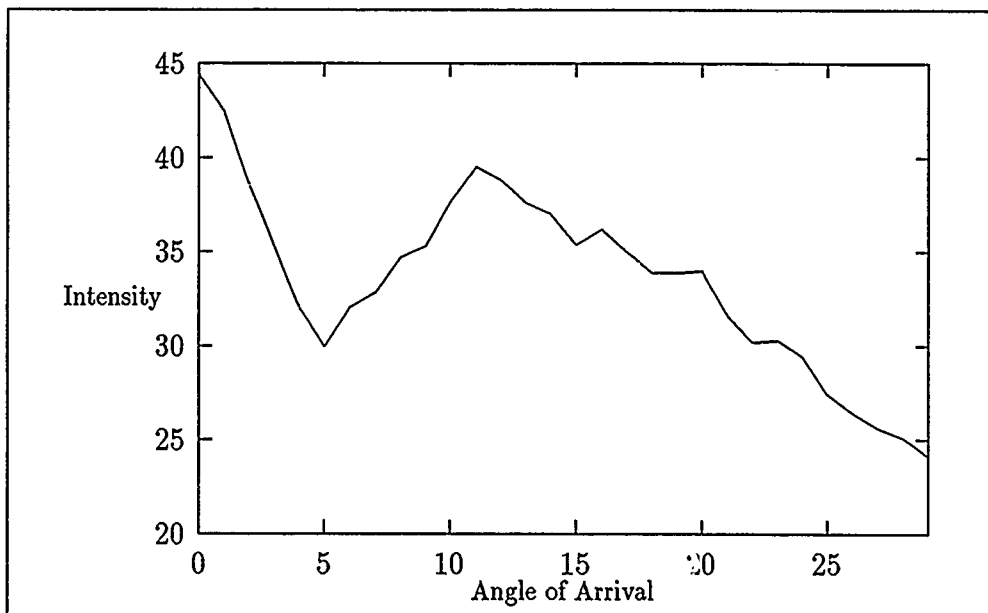


Figure 9. Intensity verse Angle of Arrival

C for the makeup of the Spiricon file). The 512-by-512 bytes of picture information is the intensity pattern out of the optical fiber. By thinking of the 512-by-512 picture file as an 262144 component array a Euclidean distance measurement could be made between each of the template picture files and the unknown picture file. The comparison with the lowest distance would then be considered the correct AOA. Although theoretically simple, the comparison of 262144 components of an unknown AOA picture with each other 262144 components of each of the templates pictures would be computationally slow.

In practical pattern recognition problems it is often useful to restrict the dimensionality of the decision space because of constraints on computational resources or available power. To solve any of these practical problems in "real-time" the number of features must be limited. (4:6)

One method of reducing the dimensionality of the decision space is by taking the Fourier transform of the pattern and then comparing the lower frequency components

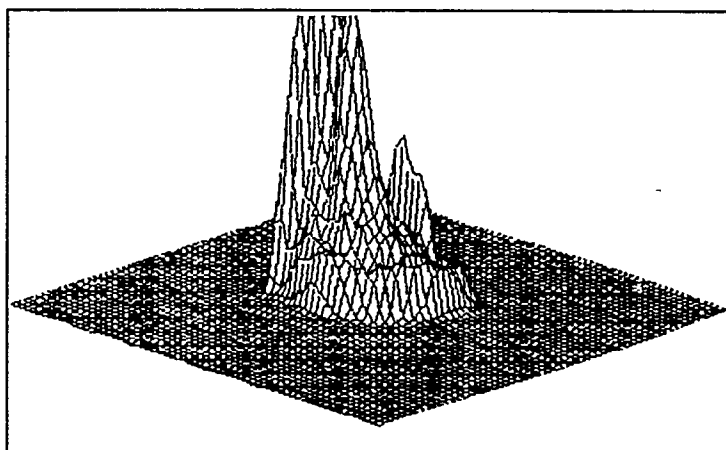


Figure 10. Before Reduction

of the transform (4:9). Capt Welker used the lower 49 components of the Fourier transform of an array representing the picture and showed that these 49 components were adequate for determining the AOA to within one degree. This research starts with the 49 component baseline.

Capt Welker also reduced down the 512-by-512 Spiricon file by pulling out the center 256-by-256 components. This was a valid reduction since the light intensity out of the fiber only excites approximately the center 200-by-200 pixels of the camera, which corresponds to the center 200-by-200 components of the Spiricon picture file (reference Appendix C). Figure 10 shows the 3-dimensional intensity plot of a complete Spiricon picture file. Figure 11 shows the plot of just the center portion of the complete file. It can be seen that no useful (at least to this method of AOA detection) information is lost, only small intensity values outside the 256-by-256 center are lost. In fact the results latter will show it is beneficial to reduce the file before Fourier transforming the file.

Figure 11 was produced by taking the original file (shown in Figure 10), pulling out the 256-by-256 center of the file, adding zeros to expand the file back to the 512-by-512 Spiricon picture size, and then putting the file back into the Spiricon file

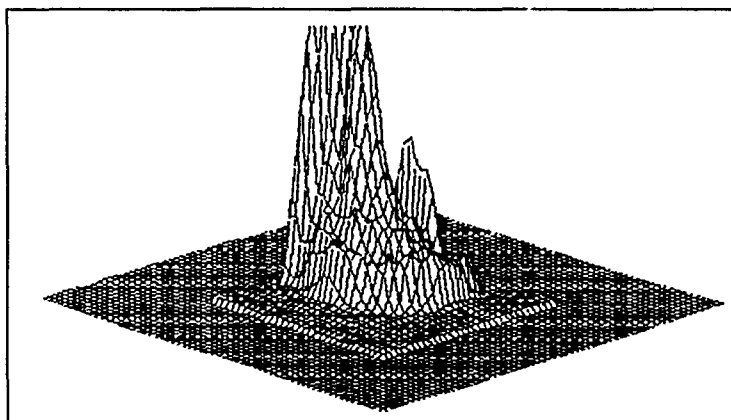


Figure 11. After Reduction

format to print out.

3.3 Laboratory Setup

The laboratory setup used for the AOA detection system used in this research is shown in Figure 12. The source laser (Helium-Neon 50mW continuous wave laser) sends its laser beam through a variable attenuator towards a beam steering instrument which angles the laser beam 90 degrees towards a collimator. The collimator expands the beam and removes high frequency noise from the laser beam. Refer to Appendix B for details on the collimator. The expanded beam goes through the neutral density (ND) filter, with an optical density of 2 (10^{-2}), and hits the front of the 1-inch-long, 3mm-diameter optical fiber mounted on the face of the camera. The optical fiber was pushed up tight against the protective glass over the camera's CID array.

The laser energy travels through the optical fiber resulting in an intensity image on the camera end of the fiber. The Spiricon software (version 5.0) running on the PC, containing the Spiricon control board, captures the resulting intensity image and stores the intensity in a Spiricon picture file. The angle at which the laser hits the optical fiber can be adjusted by moving the rotational stage to which the camera

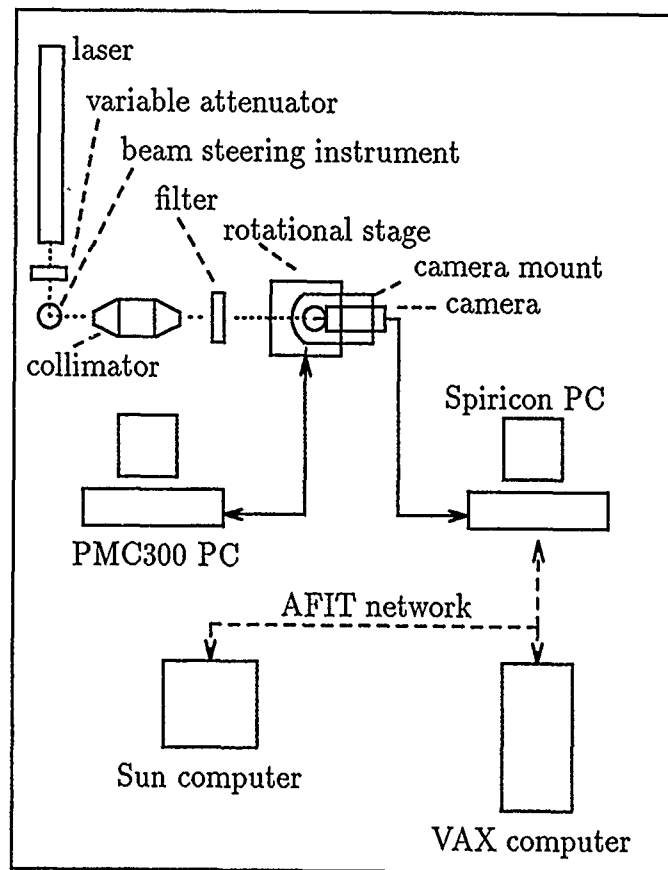


Figure 12. Laboratory Setup

is mounted. The rotational stage is controlled by the WARP software running on the PC containing the PMC300 control board.

For initial setup and alignment of the AOA detection system refer to Appendix A.

3.4 Phase I

Phase I was to baseline the research of this thesis. The baseline chosen was that of the work done by Capt Welker. Before changing any of the equipment setup various template and test pictures were taken and processed using Capt Welker's methodology of determining the AOA. The steps of the baseline methodology is as follows:

- Align system in accordance with Appendix A.
- Manually set the camera to a certain AOA and record the intensity pattern (picture).
- Repeat the above step for each of the required template angles in sequential order.
- Randomly set the camera to a certain AOA and record the intensity pattern (picture).
- Repeat the above step for each of the required test angles.
- Process the template and test picture files through the PREPROC program.
- Transfer the preprocessed picture files to the VAX computer.
- Fourier transform the preprocessed picture files using the 2DFFT program.
- Transfer the resulting transform files back to the PC.
- Run the transforms of the template files through the TEMPLATE program to create an overall template file.

- Run the transforms of the test files through the TEMPLATE program to create an overall test file.
- Run the DISTANCE program to compute the Euclidean distance to determine the correct AOAs of the test pictures.

Various template and test pictures were gathered to verify that Capt Welker's work could be reproduced. After Capt Welker's work was verified, the rotational stage was added to the optics bench setup and another set of data was taken. Template pictures were taken from 0 to 15 degrees in 1 degree increments. Template pictures (0,1,2,3,4,5,6,10,11,15,11, and 5 degrees) were taken in a random fashion. This set of data was used to first assure that the rotational stage did not change the system's ability to detect the AOA and secondly to provide a baseline set of data to see how changes to the phase one computational processing effected the detection method.

3.5 Simplifying the Methodology

Capt Welker's method required transferring the data files back and forth between the Spiricon PC and the VAX. Since the PC is limited in the size of data files it can handle at one time, this research effort wanted all of the processing accomplished on a single computer. Since the VAX could easily handle the size files required and the 2DFFT routine was already running on the VAX, the VAX seemed to be the logical choice. First the TEMPLATE and the DISTANCE routines were modified to run on the VAX leaving only the preprocessing routine still on the PC.

Since the data was cutoff at the maximum intensity level out of the camera it was theorized that the preprocessing program written by Capt Welker had little effect on the results of the AOA detection method. To test this theory a simple REDUCE program was written that pulled out the center 256-by-256 picture out of the Spiricon 512-by-512 picture. The resulting files were then run through the 2DFFT, TEMPLATE, and DISTANCE routines on the VAX computer. The end distance results were then compared to the prior results using the PREPROC program. The PREPROC

program normalized, mean filtered, and reduced the Spiricon file prior to performing a Fourier transform on the file.

The next simplifying step was to combine programs together so that multiple template and test pictures could be run through without user intervention. The first thought was to combine the programs together using a batch routine on the VAX. The trouble was that the 2DFFT routine was written to perform more than just a 2DFFT and would have to be modified to work in a batch file with the rest of the programs required. Since the 2DFFT routine was written in the ADA language and the rest of the programs were written in the 'C' language, it was determined to find a 2DFFT routine written in 'C'. The routine found was the FOURN subroutine and is given in Appendix E. Going to a 'C' 2-dimensional Fourier transform routine allowed the reduction of the number of programs required for reducing the intensity pattern into a feature vector that could be used to determine AOA.

3.6 Data Collection

Data collection required setting up two PCs. The PC containing the Spiricon system and the PC containing the PMC300 system. The PMC300 system was setup by bringing up the WARP program on the computer containing the PMC300 system. Upon turning on the PMC300 the rotational stage and software must be zeroed. First the incremental positioning routine of the WARP software was used to adjust the rotational stage to zero. Once the stage is on zero the software must be zeroed, refer to PMC300's operators manual (9:11-20). During actual data collection the absolute positioning routine of the WARP software was used to set the AOA.

The Spiricon system was setup to capture a single frame of data per AOA. The Spiricon program was started by typing cq3d from the spir directory on the Spiricon PC. Once the CQ3D program was running the system was setup to capture a single frame by the following key sequence from the main menu.

- F7 (setup menu)

- F2 (beamlink 2250 menu)
- Alt F2 (set number of frames to be capture)
- 1 (set to single frame)
- F10 (setup menu)
- F10 (main menu)

The data was collected by first setting the required AOA on the PMC300 PC and then capturing the data on the Spiricon PC by the following key sequence.

- F7 (setup menu)
- F2 (beamlink menu)
- Alt F4 (capture frame of data)
- F10 (setup menu)
- F10 (main menu)
- F4 (save menu)

The files were saved by the following conventions: %up%udeg or %up%utst. The first %u in the filename referred to the whole portion of the AOA, and the second %u referred to the fractional portion of the AOA. For example, 15p20deg referred to a template file with an AOA of 15.20 degrees. The tst file was used to designate a test file.

Once all required files were collected or the available disk space on the Spiricon PC was full the template and test files were transferred to the VAX computer for processing.

The file transfer program (FTP) was used to transfer data files over the AFIT computer network to the VAX computer. The VAX computer used was called milo. From the directory on the PC containing the Spiricon data files the following sequence was typed to send the data files to milo.

- ftp milo
- Login sequence
- binary
- mput *.*
- y for each file to transfer
- bye to end ftp session

In the above sequence the binary command was important since the Spiricon files were binary. Without the binary command the FTP program defaulted to an ASCII transfer which resulted in faulty data transfers.

Phase II required transferring the feature vectors generated on the VAX computer back to the PC and then onto the Sun computer. Due to the configuration of the AFIT network, a direct transfer from the VAX to the Sun was not possible. The reason for going to the Sun computer for phase II was because the RBF software was written to run on the Sun computer. Although the RBF software was written in 'C', the time was not available to get the software configured correctly to run on the VAX computer.

Once the required conversion program, MZNET, was run on the results from the XFM program the file containing the feature vectors for a certain series could be transferred to the Sun computer by the following sequence, using any PC computer on the AFIT network.

- ftp milo
- Login sequence
- get (filename)
- bye
- ftp thomas

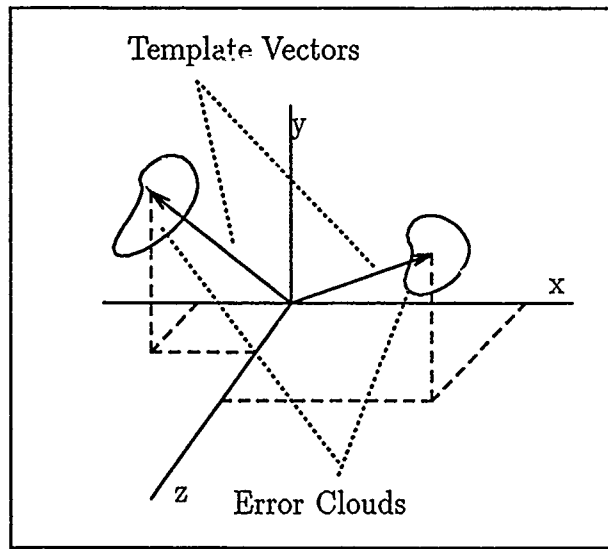


Figure 13. Error Associated with Feature Vectors

- Login sequence
- put (filename)
- bye

In the above sequence the VAX computer was called milo and the Sun computer was called thomas. Also, note that the binary command was not used because the files were ASCII.

3.7 Phase II

In this research the measurement of a class (AOA) is a 24 component feature vector of the lower harmonics of the Fourier transform of the intensity pattern out of the optical fiber. However, due to noise, no two feature vectors for the same AOA will ever be exactly alike (reference Figure 13). The question is what is the best way to set that feature vector for a given AOA so that the set feature vector (template) can be used to find all other feature vectors of the same AOA, or in reference to Figure 13 all vectors falling within the error cloud around each feature vector.

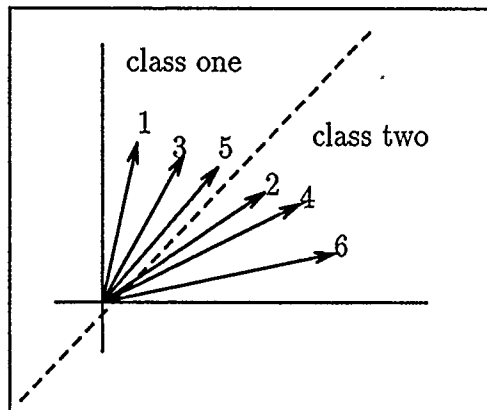


Figure 14. Two Class Example

In the first phase of this research the first intensity pattern taken was used as the template measurement. Using the first measurement as the template showed that the AOA could be determined to within $1/2$ degree using the Euclidean distance measurement technique. Since the data showed that the system only had trouble around zero and 27 degrees (that is no misclassifications between 1 and 26 degrees), it was determined to restrict the test AOAs to the area between 26 and 27 degrees to allow more data exemplars to be taken at smaller separations between AOAs. Time did not allow taking a full set of data for angles less than half a degree over the whole 27 degree range of the optical fiber system.

Restricting the AOA to two degrees allowed for many more exemplars per AOA to be taken. The question then arose what is really a template verses a test vector sense they both are taken under the same conditions. In reality none, since each feature vector varies slightly from the others of the same AOA due to noise. So how is the best template feature vector chosen? For example Figure 14 shows six feature vectors: three each of two classes or AOAs. If vectors 1 and 2 were the first vectors taken then they would be designated as the template vectors. The remaining vectors would be classified as test vectors. It can easily be seen from the example that using a Euclidean distance measurement to determine the class of vector 5 would fail and

label vector 5 as belonging to class two since the template for class two (vector 2) is closer than the template for class one (vector 1). But, it can easily be seen that if vectors 3 and 4 were chosen as the template vectors the Euclidean distance measurement method would correctly classify all of the test vectors. What is needed is a system that could learn from all of the examples give it from one class so that the system is not dependent on a single feature vector (a vector which may or may not be the best class example) as representative of a class of vectors.

3.8 Radial Basis Function

A radial basis function (RBF) system was being developed concurrently by Capt Zahirniak. Capt Zahirniak's (16) system was based on a three layer hybrid RBF artificial neural network. The system was hybrid in the sense that the hidden layer is set by the exemplars contained in the training set (unsupervised) and the output layer is trained by a matrix inversion method (supervised). The supervised output layer speeds the network's learning time because the output layer's interconnection weights are set through the matrix inversion method and do not have to be learned.

The three layer RBF network consists of an input, hidden, and output layer. The RBF architecture is shown in Figure 15. The nodes of the input layer take the input feature vector and fans the vector out to the hidden layer. Fanning the input vector into a higher level increases the ability of the network to classify the feature vector (14:461). The hidden layer consists of units made up of localized receptive fields and the output nodes are linear.

The nodes of the hidden layer consists of RBFs. A RBF is "a radially-symmetric function with a single maximum at its origin and which drops off rapidly to zero at large radii" (8:282). The origin or center of each RBF was preset to match an exemplar from the training set. That is if there were 40 training exemplars given to the net; there would be 40 RBF nodes in the hidden layer. Figure 16 shows a two-dimensional RBF.

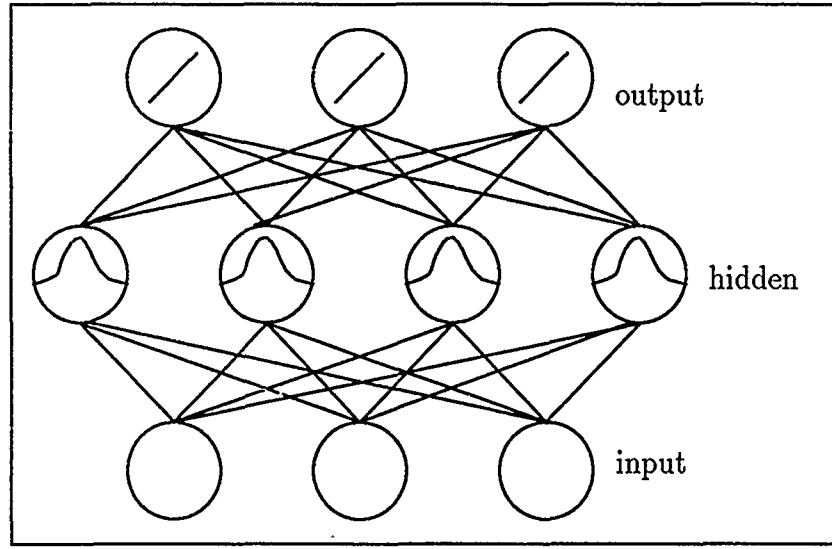


Figure 15. Radial Basis Function Network (11:3)

Since in this research the input to the network was a 24 component feature vector, the RBF actually used was 24-dimensional. Capt Zahirniak's network used the Gaussian function as the RBF for each of the nodes in the hidden layer. The output of each node in the hidden layer would be of the form:

$$H_{hp} = \Phi(\|\underline{x}_p - \underline{c}_h\|) \quad (1)$$

where the input vector \underline{x}_p is the p^{th} exemplar applied to the input layer, and vector \underline{c}_h represents the center of the h^{th} RBF node. Since Φ is a Gaussian function, Φ is defined as:

$$\Phi(\|\underline{x}_p - \underline{c}_h\|) = \exp[-\sum_n (x_n - c_{nh})^2 / 2\sigma_{nh}^2] \quad (2)$$

where n is the number of components of the input vector and σ_{nh} is the variance or the width of the h^{th} Gaussian RBF node along the n^{th} dimension (14:462-463). In this research effort the variance was initially set to four on all nodes. Then each node's width was adjusted down individually until the individual node's response to an out of class feature vector was less than the given sigma threshold value.

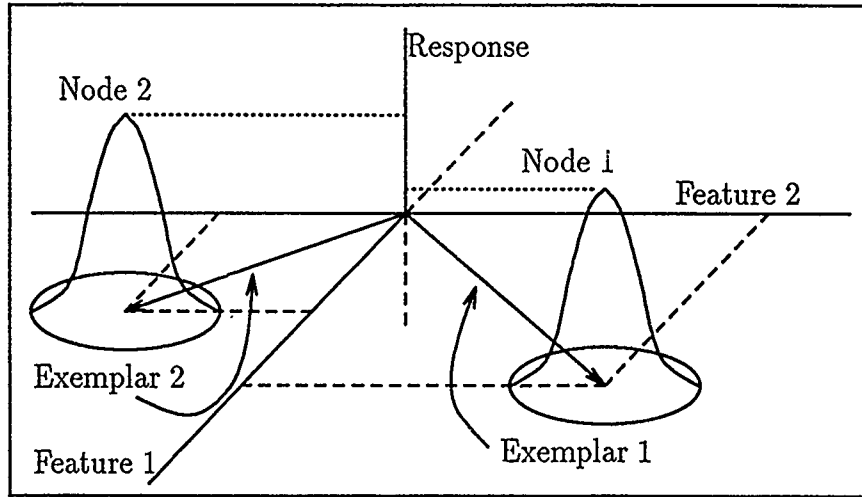


Figure 16. Example of a Two-dimensional RBF

The output of the hidden layer nodes were connected to each of the linear output layer nodes. The connection weights between the output and hidden layer were trained using the matrix inversion method (14:461).

3.8.1 Matrix Inversion The output layer of the RBF network was a linear layer connected to the hidden layer by connection weights. The connection weights could be determined by using a back-propagation algorithm or by a matrix inversion algorithm. The matrix inversion method was used and “produces the exact minimum” (14:463).

Referring to Figure 15, the input (y_{ip}) to the i^{th} output node for the p^{th} exemplar is given by the equation:

$$y_{ip} = \sum_h w_{ih} H_{hp} \quad (3)$$

where w_{ih} is the connection weight between the i^{th} output node and the h^{th} hidden layer node and H_{hp} is the output of the h^{th} hidden layer node for the p^{th} input exemplar. In the matrix inversion method the weights are selected as to minimize

the error equation:

$$E = 1/2 \sum_i \sum_p (y_{ip} - Y_{ip})^2 \quad (4)$$

where Y_{ip} is the desired output of the i^{th} output node for the p^{th} exemplar. As shown by equation 4, error (E) by definition is the summation over all exemplars (p), given to the network, of the summation of the square of the difference between the output at a given i^{th} node, of the output layer, and the desired output of the same node.

Minimizing error implies that the partial derivative of the error with respect to any weight equals zero. For example, taking the partial derivative of both sides of equation 4 with respect to the weight w_{kl} yields:

$$\partial E / \partial w_{kl} = \sum_h w_{kh} \left(\sum_p H_{hp} y_{lp} \right) - \sum_p Y_{kp} y_{lp}. \quad (5)$$

Forcing the partial derivative to zero allows the equation to be written as:

$$\sum_h w_{kh} \left(\sum_p H_{hp} H_{lp} \right) = \sum_p Y_{kp} y_{lp}. \quad (6)$$

The second term of the left side of this equation ($\sum_p H_{jp} H_{lp}$) is a correlation matrix (**M**) of the outputs of the nodes of the hidden layer. That is how much the output of any node j of the hidden layer is like the output of any node l of the hidden layer for the p^{th} input. If the other terms of equation 6 are expanded to account for all values of k and h then these terms also can be written as matrixes. The term w_{kh} would yield a weight matrix **W** as k and h are varied. The term $\sum_p Y_{kp} y_{lp}$ would also yield a matrix (**Y**) as k and h are varied. Therefore, dividing both sides of equation 6 by **M** gives the equation for the weight matrix that would yield the minimum error:

$$\mathbf{W} = \mathbf{Y} \mathbf{M}^{-1}. \quad (7)$$

Since **Y** and **M** are known, **W** can be directly computed. Once the weight matrix is computed the RBF network can classify a given input exemplar by picking the

output node with the highest value since each output node represents one of the possible classes. In the data supplied to the network there were 10 possible AOAs so there were 10 output nodes, one per AOA.

3.9 Software

Throughout this research each program change was checked for accuracy. The main program, XFM, was the result of combining five independent programs: REDUCE, FLOAT, FOURN, ORGANIZE, and TEMPLATE. All of the independent program results were checked before the programs were combined into one. The combining of the five programs into the one, XFM program, allowed multiple Spiricon template or test picture files to be processed down into a template or a test vector file in one step. The five sections of the XFM program are explained below.

The REDUCE section takes in a Spiricon file and pulls out the center of the picture information. The center contains the real intensity pattern information. To this real 256-by-256 picture information file the program adds a zero valued imaginary portion resulting in a 256-by-512 matrix. The imaginary portion is required by the FOURN routine. The FOURN routine also requires the input data to be floating-point numbers. The float section changes the Spiricon unsigned byte characters into floating-point numbers. The program FOURN is then called to perform a two-dimensional Fourier transform on the matrix. The FOURN routine writes over the original matrix with the Fourier transform of the original matrix.

The ORGANIZE section takes the magnitude of the Fourier transform and then flips the resulting 256-by-256 around so that the dc term is in the center of the matrix, and the higher order harmonic are symmetrically located about the dc term. This organization of the magnitude of the transform allows the TEMPLATE section to pull out the required harmonics to produce a template or test vector. When the program is first started the program requires the following information:

- A run number

- Whether the program is to process template or test files
- The angle of the first picture file.
- The angle of the last picture file .
- How many sub-angles per degree.

Note the input files must be sequentially numbered and each file requires about 20 seconds of run time on the VAX computer. The actual run time depends on the number of users on the VAX. The resulting output file will contain the lower 49 Fourier transform components of each of the sequential picture files processed by the program. The number of sequential picture files determines the number of template or test vectors there are in the resulting template or test file. The resulting file is labeled `template(run number).vec/.doc` for template files and `test(run number).vec/doc` for test files. The `.doc` file can be printed out. The `.vec` file is used by the NACDIS and the MZNET programs.

The NACDIS program takes a template file and a test file and computes the distance between each of the vectors of the two files. The distance between each test vector and template vector is printed out. The program also prints out the estimate AOA of the test vector based on the smallest distance between a given test vector and template vectors. The program prints out to the screen the estimated AOA for each of the test vectors. The program also prints the complete distance information to a file labeled as follows: `(test run number)_nacdis_(template run number).doc`. For example if `test1.vec` was run against `template1.vec` the resulting saved file would be named `1_nacdis_1.doc`.

The MZNET program takes in a ten class binary output file from the XFM program and produces a class labeled ASCII file out. The ASCII output file is readable by the Radial Basis Function (RBF) neural network software written by Captain Zahirniak. The program's output file is also printable, to the screen or to a printer.

3.10 Summary

This chapter provided a road map of how the research for this thesis was accomplished. The chapter started out with an overview of the light transfer characteristics of the optical fiber used in this research. Next, data reduction was discussed in relation to the producing of a feature vector from the intensity image out of the optical fiber. The third area of this chapter explained the laboratory setup used to gather the data of this research. The chapter then explained phase I of the research effort, which was to set a baseline to compare system changes to. System changes were then discussed in the following section on simplifying the AOA detection system. The simplifying section explained how the processing of the information was handled exclusively by the VAX computer. The next section went into the sequence of events required to take data for this research. The data gathered was baselined in phase I and applied to a RBF artificial neural network in phase II which was covered in the last sections of this chapter. The next chapter will show the results of the work described in this chapter.

IV. Findings

This chapter presents the findings of using the Euclidean distance measurement and the radial basis function (RBF) neural network as classifiers in an AOA detection system. The first section covers the results of repeating Capt Welker's previous AOA methodology.

The next three sections (Sections 4.2, 4.3, and 4.4) cover the results of modifications made to Capt Welker's methodology. The best of the modifications were then applied to classify fifth of a degree AOAs, and the results are presented in Section 4.5. The chapter then covers phase II results. Phase II of the research effort was to classify AOAs using the RBF neural network. The results of using the RBF neural network as a classifier are presented in Section 4.6. The work done with RBF network as a classifier gave new insight into the exemplars of the AOAs. Section 4.7 presents the results of applying this new insight to the Euclidean distance methodology for classifying one-tenth of a degree AOAs.

4.1 Baseline

The first step was to baseline this research effort to the work done, in 1989, by Capt Welker. Before changing Capt Welker's laboratory setup various template and test AOA intensity pictures were taken and processed using Capt Welker's methodology of determining the AOA. The steps of the baseline methodology are given in Chapter III. Template pictures were taken from 1 degree to 8 degrees. Test pictures were taken at 4, 5, 6, 7, and again at 4 degrees. The baseline results, as shown in Table 1, show that all of the test AOAs were correctly identified. The results showed that Capt Welker's one degree of accuracy was reproducible and gave this research a starting point on which to see if improvements to the system could be made.

Table 1. Baseline Results

Actual Angle	Identified Angle
4	4
5	5
6	6
7	7
4	4

4.2 Rotational Stage

The first improvement to the system was the addition to the laboratory setup of the rotational stage. The rotational stage allowed pinpoint accuracy in the setting of the AOA of the laser beam on the front of the optical fiber. The stage's accuracy of 0.001 degree removed from the AOA detection system the inaccuracies of manually setting the AOA. The stage allowed the AOA classification methodology to be the limiting factor verses being limited by how well the AOA could be manually set and reset to various angles.

To insure that the rotational stage was working properly and that adding the stage to the laboratory setup did not have adverse effects on the detection system, another set of data was taken. Template pictures were taken from 0 to 15 degrees in one degree increments. Then test pictures (0,1,2,3,4,5,6,10,11,15,11, and 5 degrees) were taken in a random fashion. Again the data was processed using the baseline processing methodology and perfect results were obtained (all AOAs correctly identified), showing that the rotational stage did not change the system's ability to detect the AOA.

4.3 Processing Simplification

The baseline processing methodology, in order to process and classify Spiricon intensity pictures, required the following programs: PREPROC, 2DFFT (written in

Ada), PROCDATA, and DISTANCE. This baseline processing method was slow due to four main reasons: 1) the data had to be transferred twice over the AFIT network, 2) the Ada 2DFFT routine required each picture file to be processed individually, 3) the PC could not handle the size files required as efficiently as the VAX computer, and 4) unnecessary preprocessing of the Spiricon data files. In order to speed up processing time for phase I, this research only used the PC to gather Spiricon picture files; the files were then transferred over the AFIT network to the VAX computer for processing and classification.

Various steps were taken to assure that changes to the baseline processing method did not alter the reliability of the system to determine AOA. The first step of changing over to a 'C' routine to take the 2DFFT was checked. Initially the VAX constantly did memory dumps when processing the results of the 'C' 2DFFT (subroutine FOURN). It was found that the results were causing arithmetic overflows in the Euclidean distance routine. The problem turned out to be incorrect data transfers to the FOURN subroutine. The FOURN subroutine required the data to be passed as an array and initially the data was being transferred as a matrix resulting in faulty Fourier transform results. Once the transfer problem was corrected, the initial results from the FOURN subroutine were extremely high and needed to be scaled by a factor of 512^2 to give results comparable to the Ada 2DFFT routine results. Next the DISTANCE routine was modified and tested on the VAX and showed that the VAX DISTANCE routine gave the same results as the PC version. The final step was to assure that leaving out the PREPROC program did not degrade the AOA detection system.

Three methodologies were compared: 1) processing the data with PREPROC, 2) processing just the reduced 256^2 byte data field of the Spiricon picture file, and 3) processing the entire 512^2 byte data field of the Spiricon picture file. All three methods were able to determine the one degree test angles of arrival. But, did any of the three methods show more promise of determining smaller increments of AOA?

Table 2. Comparison of Processing Methods

By Distance to Next Nearest Neighbor			
Angle	Procdata	256x256 Spiricon	512x512 Spiricon
0	4.8	4.7	1.4
1	6.6	6.9	2.5
2	11.8	11.9	3.9
3	12.6	12.8	3.4
4	9.2	10.1	2.4
5	9.8	9.7	2.5
6	8.8	8.9	2.8

To answer this question a figure of merit (FOM) was chosen. The FOM chosen was the distance to the next nearest neighbor to the test vector. The nearest neighbor to the test vector was always the corresponding template vector. The next nearest neighbor, in all methods, was the template vector representing a plus or minus one degree from the correct angle.

Table 2 shows the distance, for each of the three methods, to the next nearest neighbor for the angles 0 to 6 degrees. Table 2 lists three columns. Column one gives the results using the PREPROC program. Column two gives the results using only the center 256² byte center of the Spiricon picture file. Column three gives the results of using the entire 512² byte data field of the Spiricon picture file. It can be seen from Table 2 that the PREPROC program does not significantly improve the distance to the next nearest neighbor, in fact in 4 out of the 7 angles the distance is actually less than that of the next nearest neighbor using just the REDUCE routine. It can also be seen from Table 2 that using the entire Spiricon data field significantly decreases the distance to the next nearest neighbor.

In pattern recognition the greater the distance between feature vectors the easier it is to identify the feature vectors; therefore, the method using the entire Spiricon picture file was not continued. Since just using the reduced version was just

Table 3. Results 49 Component Feature Vector

Missed $\frac{1}{2}$ AOAs	
Actual AOA	Indicated AOA
0.5	1.0
6.5	6.0
18.5	18.0
19.5	19.0
22.5	22.0
23.5	23.0
26.0	25.5
26.5	26.0
27.5	27.0
29.5	29.0

as good if not better than the PREPROC, the REDUCE method was selected for this research. Not only did the REDUCE method give the best overall FOMs, but it also allowed for quicker processing time over the other two methods.

Since the AOA detection method showed that detection of the AOA to one degree was possible, the next step was to reduce the data gathered to angles of less than one degree. A set of Spiricon picture files were taken covering every half degree from 0.0 to 29.5 degrees. The results using the Euclidean distance methodology with the 49 component feature vector resulted an accuracy of 83 percent or 10 misses out of the 60 test vectors. The angles missed are showed in Table 3. The results show that the system was accurate to within one-half plus or minus a half degree.

The 49 component feature vector was the dc and the lower three harmonic components of the Fourier transform of the intensity picture out of the optical fiber for a given AOA. In actuality the magnitude of the harmonic components are being used; therefore, the feature vector is composed of symmetrical data about the dc component, as shown in Figure 17. Which means there are two sets of identical 24 components in each 49 component feature vector. Since the data is identical why

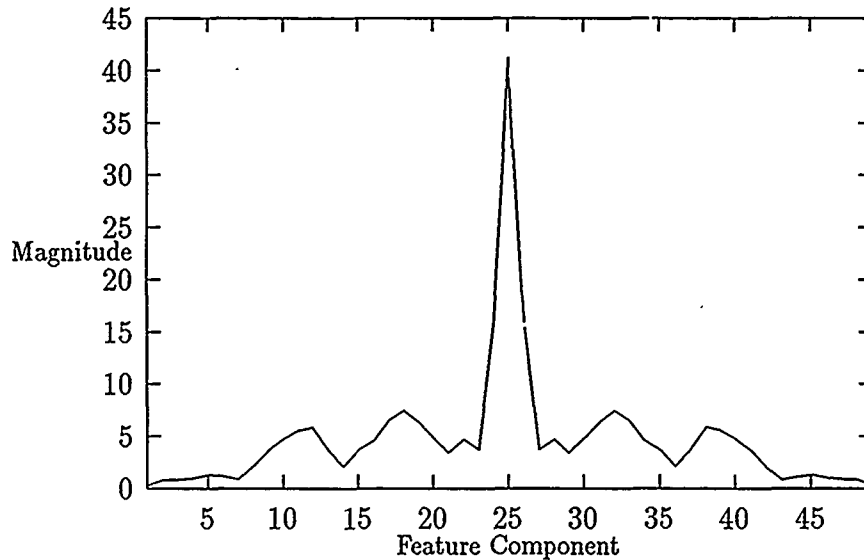


Figure 17. Magnitude Plot of a Typical Fourier Transform

keep it?

The Euclidean distance computation was modified so that it would compute the Euclidean distance on just 25 components. Table 4 shows the angles missed using the 25 component feature vector. Table 4 shows that the miss rate increased by the deletion of the repeated data. Not only did the detection system miss all of the angles missed using 49 component feature vector, but the system also missed three more angles resulting in a lower classification rate of only 78 percent. This result is not unexpected when looking at the components of the feature vector.

Referring to Figure 17 it can be seen that the typical dc component can be two to forty times greater than the other components of the feature vector; therefore, throwing out half of the symmetrical data just increases the dc components effect on the results of the distance measurement. The level of the dc component washes out the smaller component levels of the lower three harmonics.

Since the dc component dominates the Euclidean distance measurement, what would happen if the dc component was disregarded? Table 5 shows the results of just

Table 4. Results 25 Component Feature Vector

Missed $\frac{1}{2}$ AOAs	
Actual AOA	Indicated AOA
0.5	1.0
6.5	6.0
9.5	10.0
10.5	11.0
18.5	18.0
19.5	19.0
22.5	22.0
23.5	23.0
24.5	24.0
26.0	25.5
26.5	26.0
27.5	27.0
29.5	29.0

Table 5. Results 24 Component Feature Vector

Missed $\frac{1}{2}$ AOAs	
Actual AOA	Indicated AOA
0.5	1.0

using a 24 component feature vector (no dc, just the lower 3 harmonic components). Table 5 shows that using only 24 components greatly decreases the error rate. The detection system only missed one angle giving a detection rate of 98 percent.

While all three feature vectors used (49, 25, and 24) allowed classification of AOA to within one-half plus or minus half of a degree, the 24 component feature vector seemed to have the best classification ability.

4.4 Quarter Degree Results

Decreasing the AOA to quarter degree steps required significant number of measurements to be taken; therefore not all angles were tested. Three sample sections were chosen (5.0 to 6.75 degrees, 15.0 to 16.75 degrees, and 25.0 to 26.75 degrees) to determine the ability of the system to detect quarter degree AOAs. For the angles tested, Table 6 indicates that the system can classify angles of arrival as low as a quarter of a degree when using the 24 component feature vector. Again the 24 component feature vector out performed the 25 and the 49 component; therefore, the 25 and the 49 component vectors were not tested further in this research.

4.5 Fifth of a Degree Results

Decreasing the incremental step between AOAs to a fifth degree again required to many measurements to test the entire FOV of the optical fiber. Also many repeated measurements were desired, to see if the system stayed stable through many measurements. The area between 26.0 and 27.80 degrees was chosen because in the quarter degree tests this area showed a weakness in the system's ability to classify the AOA (at least with the 25 and 49 component feature vectors).

Table 7 shows multiple fifth of a degree data sets compared against each other. Five data sets were taken for the test angles 26.0 to 27.8 degrees. The data sets were labeled 30 through 34 referring to the run numbers assigned to the data sets when the data was processed through the XFM program. The first column of the table shows which data set was being used as the template set. Each row shows the accuracy rate of a given data set when it was being used as a template, and a Euclidean distance measurement was made to each of the remaining data sets. Table 7 shows the fallacy of always choosing the first data set as the template set. In the 30 series data the first data set (run 30) turned out to be the poorest template.

Another data set (series 50) was taken to see if the results would be comparable to the series 30 data. The series 50 data, as shown in Table 8, showed the that the

Table 6. Quarter Degree Results

Actual AOA	Indicated AOA per # of components		
	49	25	24
5.0	5.0	5.0	5.0
5.25	5.25	5.25	5.25
5.50	5.50	5.50	5.50
5.75	5.75	5.75	5.75
6.0	6.0	6.0	6.0
6.25	6.25	6.25	6.25
6.50	6.50	6.50	6.50
6.75	6.75	6.75	6.75
15.0	15.0	15.0	15.0
15.25	15.25	15.25	15.25
15.50	15.50	15.50	15.5
15.75	15.75	15.75	15.75
16.0	16.0	16.0	16.0
16.25	16.25	16.25	16.25
16.50	16.50	16.50	16.50
16.75	16.75	16.75	16.75
25.0	25.0	25.0	25.0
25.25	25.25	25.25	25.25
25.50	25.50	25.50	25.50
25.75	25.75	26.0	25.75
26.0	26.0	26.0	26.0
26.25	26.0	26.0	26.25
26.50	26.50	26.50	26.50
26.75	26.75	26.75	26.75
# Wrong	1	2	0
Error (deg)	0.25	0.25	0
% Right	90	80	100

Table 7. Series 30 (0.2deg) Results

Euclidean Distance Methodology					
Template	Test				
	30	31	32	33	34
30	-	100%	40%	50%	60%
31	50%	-	100%	100%	100%
32	50%	100%	-	100%	100%
33	50%	100%	100%	-	100%
34	90%	100%	100%	100%	-
All Errors $\pm 0.2\text{deg}$					

Table 8. Series 50 (0.2deg) Results

Euclidean Distance Methodology				
Template	Test			
	50	51	52	53
50	-	80%	30%	60%
51	80%	-	50%	60%
52	30%	40%	-	60%
53	60%	70%	50%	-
Max Error $\pm 0.4\text{deg}$				

Table 9. Series 90 (0.2deg) Results

Euclidean Distance Methodology				
Template	Test			
	90	91	92	93
90	—	80%	90%	80%
91	80%	—	100%	100%
92	80%	90%	—	90%
93	100%	90%	90%	—
Max Error $\pm 0.2\text{deg}$				

system could get out of alignment. Upon going back and checking the alignment of the system it was found to be out of alignment. The collimator was no longer aligned parallel to the beam which resulted in a lower intensity level into the front of the optical fiber.

The system was realigned and series 90 data was taken. The results of the series 90 runs are shown in Table 9. Just before taking the data, between each data set, and after the last data set was taken the camera was zeroed and the zero degree intensity picture was compared to that shown in Appendix A to assure that the system was still in alignment. This reverifies the system's ability to determine the AOA to one-fifth plus or minus a fifth of a degree when the system is stable. As can be seen from series 30 and 90 runs, selection of the template set determines how well the system performs. Which leads to phase II to see if the RBF network would learn from all the data given it to increase its accuracy at determining the AOA.

4.6 Phase II Results

Capt Zahirniak's RBF neural network had many options. Initially the RBF network gave accuracies around 90 percent. Which could be matched by the Euclidean distance methodology if the right run, like run 34, was used as the template.

The greatest improvement in the RBF neural network's ability to determent

the correct AOA came when Capt Zahirniak suggested normalizing the feature vector into the RBF neural network. Using the normalization option of the RBF network software immediately improved the results of the series 50 runs. Prior Euclidean distance results with the 50 series gave errors as high as 0.4deg with over half of the AOAs miss classified. Normalizing the 24 component feature vector into the RBF network immediately jumped the accuracy up to 100 percent.

Initially the RBF network was setup to take the first 30 or 40 exemplars (depending on the number of runs in a series) as the training exemplars and then take the last ten exemplars of the run as the test exemplars. This gave excellent results; but, this required the first exemplars of the series to contain the best template exemplars. To avoid this favoritism and make the results more meaningful, the randomizing option of the RBF network was used.

Selecting the randomizing option allowed the network to pick various exemplars from each class to train the network. By varying the data seed the network software would randomly select a different group of training exemplars from the data file. As expected, the accuracy of the network varied (approximately 10 percent) as different exemplars were chosen.

However, it was also found that the accuracy could be brought back up to 100 percent by decreasing the sigma interference factor of the RBF network. The RBF network software would automatically keep scaling down the size of individual RBFs until they would respond with a value less than the interference factor. For example if the sigma interference factor was set to 0.5, a given RBF node's width would be scaled down until the node's output was less than 0.5 for an adjacent out of class training exemplar.

The results of running the 0.2deg data through the RBF network are given in Tables 10. Ten different data seeds were used to give various input exemplars to the RBF network to train the network. The sigma threshold was set to 0.5 unless the network failed to correctly classify the test exemplars. In those two cases, using data

Table 10. RBF 0.2deg Data Results

RBF Network Methodology		
Series	Data Seed	% Correct
30	1-10	100%
50	1-10	100%
90	1-10	100%

Table 11. RBF 0.1deg Data Results

RBF Network Methodology		
Series	Data Seed	% Correct
200	1-10	100%
Sigma Threshold = 0.1		

seed four in series 30 and using data seed five in the series 50, the sigma threshold was decreased to 0.1 to give perfect results.

4.6.1 RBF and 0.1deg AOA Increments Since the results using the RBF network to classify AOAs to one-fifth of a degree was 100 percent; the next step was to try the network on one-tenth degree AOAs. A set of data was taken in one-tenth of a degree increments between 27.0 and 27.9 degrees. The results of the classification ability of the RBF network are shown in Table 11. In order to correctly classify the AOAs a sigma threshold of 0.1 was required. A few tests were conducted with the one-tenth of a degree data to see what was the effect of decreasing the number of training exemplars on the network's ability to classify test exemplars. Table 12 shows that as the number of training exemplars decreases so does the RBF network's ability to classify.

Table 12. Accuracy vs Varying # of Training Exemplars

RBF Network Methodology			
Data Seed	# of Training Sets		
	1	2	3
1	90%	93.3%	100%
2	77.5%	96.7%	100%
3	90%	96.7%	100%
4	80%	93.3%	95%
5	75%	90%	95%
Sigma Threshold = 0.1			

Table 13. Normalized Series 30 (0.2deg) Results

Euclidean Distance Methodology					
Template	Test				
	30	31	32	33	34
30	—	100%	100%	100%	100%
31	100%	—	100%	100%	100%
32	100%	100%	—	100%	100%
33	100%	100%	100%	—	100%
34	100%	100%	100%	100%	—

4.7 Normalized Data

After seeing the improved results with the RBF neural network after normalizing the results of the Fourier transform, it was decided to go back and normalized the data into the Euclidean distance methodology.

Table 13 shows that normalizing the data increased the accuracy rate to 100 percent and removed the ambiguity of using run 30 as a template.

Table 14 shows the results of normalizing the series 50 run. Normalizing the data in the series 50 runs brought the accuracy rate up to 100 percent for most templates and brought the totally unclassifiable (at least unclassifiable previously by the Euclidean distance methodology) data to a worst case error of plus one-fifth

Table 14. Normalized Series 50 (0.2deg) Results

Euclidean Distance Methodology				
Template	Test			
	50	51	52	53
50	-	100%	100%	100%
51	*90%	-	100%	100%
52	100%	100%	-	100%
53	100%	100%	100%	-
* 1 miss of +0.2deg				

Table 15. Normalized Series 90 (0.2deg) Results

Euclidean Distance Methodology				
Template	Test			
	90	91	92	93
90	-	100%	100%	100%
91	100%	-	100%	100%
92	100%	100%	-	100%
93	100%	100%	100%	-

of a degree.

Table 15 shows that normalizing the series 90 data increased the accuracy from the 90 and 80's percentile to a perfect 100 percentile run.

One series's accuracy decreased when the input feature vectors were normalized. The half-angle series actually decreased in accuracy from 98 percent to 96 percent due to the miss-classification of one more angle than with the non-normalized data. The non-normalized run (reference Table 5) only missed the 0.5 degree AOA where as the normalized run missed AOAs of 0.5 and 1.0 degrees. The miss on both AOAs was plus one-half degree.

Table 16. Non-Normalized 0.1deg AOA Results

Euclidean Distance Methodology					
Template	Test				
	220	221	222	223	224
220	-	40%	40%	10%	50%
221	40%	-	20%	10%	30%
222	30%	20%	-	30%	60%
223	20%	10%	20%	-	20%
224	60%	40%	80%	70%	-
worse case error +0.6deg					

Table 17. Normalized 0.1deg AOA Results

Euclidean Distance Methodology					
Template	Test				
	220	221	222	223	224
220	-	100%	100%	90%	100%
221	100%	-	100%	80%	100%
222	100%	100%	-	100%	100%
223	90%	90%	100%	-	80%
224	90%	90%	80%	80%	-
all errors $\pm 0.1deg$					

4.8 One Tenth of a Degree AOA

Originally one fifth of a degree was thought to be the limit because early attempts to detect lower AOA's proved haphazard at best, refer to Table 16. But normalizing the data, as shown in Table 17, brings the previous data up in some cases as high as 80 percent over previous attempts to classify the AOA data using the Euclidean distance methodology. The non-normalized data had missed AOAs as high as plus 0.6deg. The normalized data results had misses of only plus or minus 0.1deg and if the 222 run was used as the template the accuracy rate for classifying the exemplars was a 100 percent.

4.9 Summary

This chapter presented the findings of using the Euclidean distance measurement and the RBF neural network as classifiers in an AOA detection system. The first sections (Sections 4.1 through 4.5) showed that Capt Welker's previous AOA classification work was repeatable and showed the results of improvements made to the system. Improvements such as the automated rotational stage and the reduction in the feature vector which allowed the Euclidean distance methodology to classify AOAs as low as one-fifth of a degree. The next section (Section 4.6) gave the results of using the RBF neural network to classify AOAs. The section showed how decreasing the sigma threshold value and normalizing the feature vector allowed classifying AOAs down to one-tenth of a degree with a hundred percent accuracy. The final data presentation section (Section 4.7) goes back to the Euclidean distance methodology and with the insight of normalizing the input feature vector, gained from the RBF section, showed how normalizing the data improved the results. Improvements as high as eighty percent as was shown in the one-tenth of a degree data.

V. Conclusions and Recommendations

This chapter presents the conclusions that can be drawn from this research effort and presents some areas in which this research could be expanded. The main accomplishment of this research effort was to setup a AOA detection system that could classify AOAs to one-tenth of a degree. This is an improvement of ten times the previous work using a similar laboratory setup.

5.1 Baseline Conclusions

The baseline system of this research was that of Capt Welker's AOA detection system which showed that the AOA of incident laser energy on the front of an optical fiber could be determined to within one degree. The baseline system lack of resolution greater than one degree was mainly due to the lack of repeatability of intensity measurements. Since each AOA was set by hand, angular resolution of the baseline system was limited to approximately one degree. This lack of repeatability of an intensity measurement was overcome in this research by the addition of the automated rotational stage. The rotational stage, due to its accuracy of 0.001 degree, gave the baseline system repeatability of measurement and allowed the reproducing of Capt Welker's one degree of AOA classification.

5.2 Phase I Conclusions

Phase I of this research effort centered around the Euclidean distance methodology to classify AOAs. Euclidean distance measurements were made between various template and test feature vectors. The smallest distance between a template feature vector and that of the test feature vector classified the test vector.

The feature vector consisted of magnitude components of the lower harmonics of the Fourier transform of the intensity pattern out of the optical fiber due to the incident laser energy on the front of the optical fiber. This research concluded that

using only the lower three harmonics with no dc term made the best feature vector. Using a 24 component feature vector allowed the system to achieve a 100 percent accuracy on quarter degree AOAs. Where as only 80 percent accuracy was obtained when the dc component was included. Also, normalizing the 24 component feature vector before computing the Euclidean distance allowed the classification of AOAs down to $0.1 \pm 0.1\text{deg}$.

The normalizing of the feature vectors removed small intensity variations from the data. This conclusion can be drawn from the 50 series data. Initial results of performing classification on the 50 series data was dismal, and it was found that the intensity level out of the collimator had dropped from the intensity level noted during the 30 series run, due to the collimator falling out of alignment with the laser's path on the optics bench. Without normalization the 50 series results showed correctly classified AOAs as low as 30 percent and angular errors as high as $\pm 0.4\text{deg}$. With normalization of the feature vectors the lowest accuracy was a 90 percent with one miss of $+0.2\text{deg}$ when using run 51 as the template file. All other runs of the 50 series gave perfect results when used as the template file.

Without normalization one-fifth of a degree accuracy was the limit with the Euclidean distance methodology and feature vector used in this research. Without normalization one-tenth of degree results were worthless. With normalization the Euclidean distance methodology could classify the AOAs down to $0.1 \pm 0.1\text{deg}$. One run (the 222 file) classified all test runs perfectly indicating that picking the right template file is very important to the results in the Euclidean distance methodology.

5.3 Phase II Conclusions

Phase II of this research centered around using a radial basis function (RBF) neural network as an AOA classifier. Using the same 24 component feature vector from phase I and various template exemplars from each series, the RBF could classify AOAs to 0.1 degree with a 100 percent accuracy. The Euclidean distance

methodology could only match the RBF network's accuracy if the correct run of a given series was chosen as the template. For example, in the 220 series the Euclidean distance methodology was only a hundred percent when the 222 run was used as the template. The ability of the RBF network to use more than one run of a series as a template allowed the RBF to automatically gain in accuracy over the Euclidean distance methodology of phase I.

5.4 Recommendations for Further Research

Further automation of the data gathering portion of this research is needed. In this research two PCs were required. One PC ran the Spiricon software to gather the intensity measurements out of the optical fiber, and the other PC ran the WARP program to control the automated rotational stage. What is desired is a system controlled by a single program with the ability to measure the intensity profile of an AOA, move to the next AOA, and repeat the sequence until all required AOAs are measured without operator intervention.

The rotational stage does provide 'C' source code so that a control program could be written to control the rotational stage instead of using the WARP program. The present Spiricon system does not give the user the option of writing control code. What is needed is a system like the Spiricon system, but one that allows the user to write 'C' control code for it. Then a single data gathering program could be written to allow many small AOAs to be repeatedly measured.

A totally automated system could be configured if the present Spiricon system was to be replaced with the Spiricon LBA-100 laser beam analyzer unit. The LBA-100 is a stand-alone unit capable of being controlled though a serial port on a PC computer. So if the LBA-100 was connected to a serial port on the PC containing the PMC-300 control board, then a single 'C' program could be written to control the setting of the AOA and capturing of the associated AOA intensity pattern.

In this research, each measurement required approximately five minuets, most

of which was use by the Spiricon software to store the intensity image. Also due to only having ten megabytes free on the hard disk for storage required the measurement process to be stopped periodically to move data over to the VAX so as to allow more data to be taken. A minimum of a 80 megabyte hard drive is required for an automated system. The 80 megabytes would allow approximately 300 intensity images to be stored. Three hundred images could handle the entire FOV of the optical fiber with an angle spacing of 0.1deg.

Along with the automation of data gathering goes the requirement to have the laboratory setup in an area that can be kept dark during data gathering. This could be accomplished by moving the optics bench to its own room or enclosing the bench within an enclosure (such as, Newport's TE2406 lightweight table enclosure).

This research shows that AOA can be determined down to 0.5 degree over the entire FOV of a 3mm-diameter optical fiber. The next research needs to prove that the classification accuracy of 0.1 degree, as seen from 27.0-27.9deg can be shown to hold true over the entire 26 degree FOV of the optical fiber. To accomplish this accuracy of 0.1deg will require approximately 1300 images to be taken (26deg times 10deg/AOA time 5 repetitions). The five repetitions are required to give the RBF neural network the accuracy down to 0.1deg. With 1300 measurements it is easy to see why a totally automated data gathering system is recommended for future research.

Along with the area stated above, the following areas should also be investigated:

- Increasing FOV of the AOA detection system through the addition of more optical fibers to form a compound eye type of system.
- Using a non-linear filter to allow the system to handle varying laser beam intensities.

- Performing the Fourier transform in optics; verses the using of a computer routine.
- Check system accuracy with different laser sources.

Hopefully the AOA detection accuracy of 0.1deg will provide incentive for the continuation of research into AOA detection using the intensity patterns out of an optical fiber.

Appendix A. Alignment Procedure

The purpose of the following alignment procedure is to initially setup the angle of arrival detection system. The first section explains how to build an alignment tool. The next section explains how to assure that the laser beam is parallel to the top of the optics bench and centered on the optical fiber mounted on the front of the camera. The final section explains how the initial power level is set.

A.1 Alignment Tool

To aid in aligning the system a simple alignment tool was built out of some common optics equipment and a clear plastic square. The following items were used:

- Clear plastic square (10in)
- Optical rail
- Adapter plate with mounting screws
- Two long screws
- White correction fluid

Assemble the tool as shown in Figure 18. The following steps will align the tool to the center of the optical fiber.

- Remove the collimator from its mounting base.
- Remove the neutral density filters.
- *Caution* never look into the laser's path and be careful of laser reflections off of metal surfaces!
- Turn on the laser.

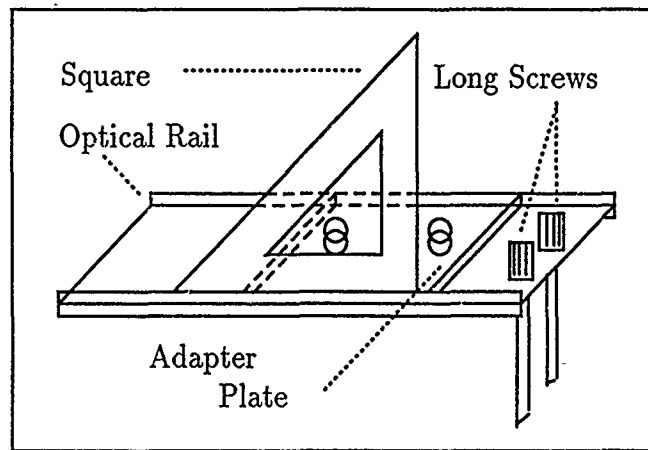


Figure 18. Alignment Tool

- Adjust the laser beam by using the adjustments on the beam steering instrument so that the beam hits the front of the optical fiber dead on (note the camera should be set to zero degrees).
- Place the alignment tool in front of the camera and square it up.

The alignment tool is squared up by forcing the two protruding screws against the edge of the optics bench. For good alignment, it is important that the vertical section of the clear plastic square be as close to the edge of the rotational stage as possible. Mark the point at which the laser beam passes through the square with a small dot (approximately the size of the laser beam) of white correction fluid.

The alignment tool is now aligned to the center of the optical fiber mounted on the camera.

A.2 Laser Beam Alignment

The laser beam must be parallel to the top of the optics bench. The direction of the laser beam is controlled by the beam steering instrument. The following steps are to align the beam. Move the alignment tool up close to the beam steering equipment. Now there are two points (the dot on the alignment tool and the center of the fiber on the camera) in line on the optics bench. Adjust the controls on the beam steering equipment until the beam illuminates the white dot brightly and the center of the fiber on the camera. Next the collimator must be aligned to the laser beam. Refer to the laser collimator instruction manual for this alignment (10:10-13).

A.3 Attenuator Adjustment

The final step in the alignment process is to adjust the variable attenuator for approximately -30dBm of power into the fiber.

- Place the neutral density filters (optical density of 2) between the collimator and the camera and in line with the laser beam.
- Place the power meter probe in front of the camera and assure that it is placed for maximum dBm reading.
- Adjust the variable attenuator until the power meter reads approximately -30dbm.
- Remove the power meter's probe from the laser's path.

The system is now aligned and should give a zero degree picture on the Spiricon monitor with no visible areas of saturation. There will be some pixels saturated randomly through out the pattern.

A.4 Zero Degree Baseline

After the above alignment has been accomplished the zero degree intensity pattern, as shown on the Spiricon's real time monitor, should be similar to the

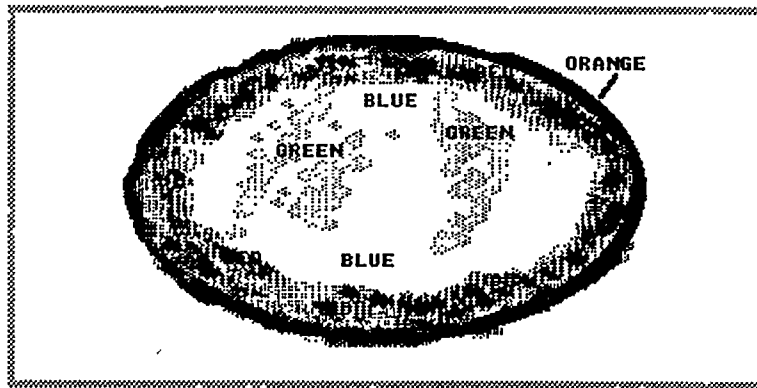


Figure 19. Zero Degree Intensity Pattern



Figure 20. Zero Degree Intensity Slice

pattern shown in Figure 19. (Note: this pattern was based on the default Spiricon color scheme.) The intensity pattern shows a thin ring of medium intensity (orange), followed by a higher intensity ring (red), dropping off to blue and green intensity levels in the center. Varying center intensity levels were observed at the zero degree position of the optical fiber.

A sliced view of the intensity pattern represented in Figure 19 is shown in Figure 20. The picture shown in Figure 20 was made by the Spiricon system after the above alignment was accomplished and with the AOA at zero degrees.

Appendix B. Collimator

The laboratory setup was devised by Captain Welker to simulate a laser source located on the ground hitting a sensor located on a satellite. Since a laser traveling from the ground to a satellite would spread out and appear to the sensor as a uniform amplitude plane wave, a collimator is required in the setup.

The collimator expands the beam so that the beam hitting the optical fiber (sensor in this research) is approximately a uniform amplitude plane wave. The output of the collimator is a 2.5cm diameter Gaussian profile intensity beam. Also, collimator provides for removing extraneous noise from the beam.

Extraneous noise is filtered out from the beam by optically Fourier transforming the incoming laser beam, only allowing the low frequency components of the Fourier transform to pass, and then inverse Fourier transforming the beam. This filtering is accomplished by having two lenses with a common focal point between the lenses. At the common focal point a 10 micron pinhole is placed which performs spatial filtering of the beam. Noise energy (such as dust on the beam steering instrument's mirrors) will usually fall outside of the pinhole resulting in only a smooth beam passing through the pinhole and out of the collimator (10:3-4). Refer to Figure 21.

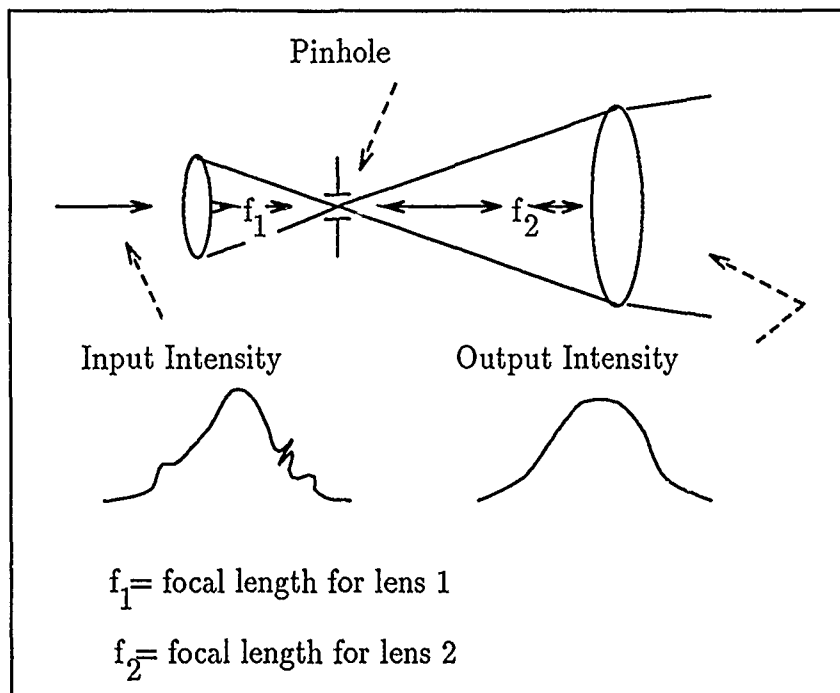


Figure 21. Collimator

Appendix C. Spiricon File

The output of the Spiricon system is a 263,052 unsigned byte (8 bits) file per picture. Figure 22 shows the layout of a Spiricon file. The break down of the file is as follows:

- 2 bytes of Spiricon header (3039 hex)
- 262144 unsigned bytes of picture information representing the 512-by-512 pixel output of the camera
- 906 bytes of Spiricon software data

The first two bytes of a Spiricon file contains a header. The header signifies to the Spiricon software that the following file is a Spiricon file.

The next 262144 bytes contain the picture information. Each byte of picture information represents the intensity of light hitting the corresponding pixel element of the camera. The value of each pixel can range from 0 to 255. The pixel elements are spaced 0.015mm apart. The fiber is 3mm in diameter; so, the intensity pattern out of the fiber excites approximately 200-by-200 of the pixels in the center of the camera's 512-by-512 pixel array. Therefore the Spiricon file can be reduced to a 256-by-256 file without losing any information. In fact decreasing the file size also decreases the noise into the system and aids in the AOA recognition method.

The last 906 bytes of the Spiricon file gives an area to store file information. The area can be used to store information such as the owner of the Spiricon software, date, time and other information pertaining to the setup of the Spiricon software at the time of picture storage. Sixty-eight characters of this 906 bytes is user definable.

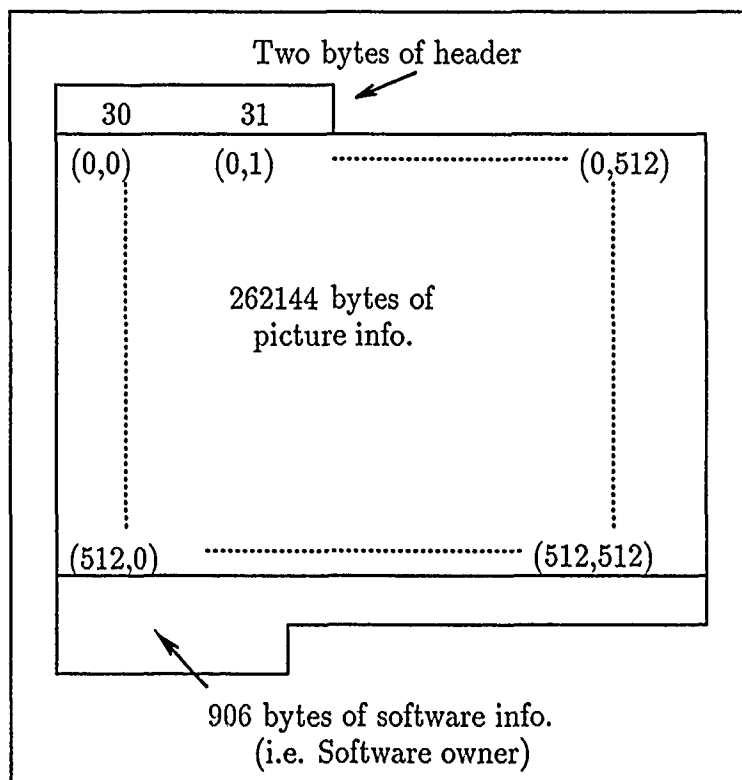


Figure 22. Spiricon File

Appendix D. Program XFM

This appendix provides the 'C' source code for the XFM program. The XFM program takes in sequentially numbered Spiricon files, reduces the files down, floats the data of each file, performs a Fourier transform on each file, and puts the dc component and the lower three harmonics of each transform in two files.

The two files allow for further processing of the data and for the viewing or printing of the lower three harmonics of each of the input files. The directly viewable (as in on screen viewing) or printable file ASCII has the suffix *.doc*. The other file has a suffix *.vec* which stands for feature vector file. The feature vector file is readable by the NACDIS program (refer to Appendix F) and by the MZNET program (refer to Appendix G). The files are also labeled to as which type of data was read in and what run number the operator wishes to assign to the data file.

There are two types of sequentially numbered data files the XFM program will accept: template or test files. Both files are physically the same; but, allow the operator to create template or test files to submit to the NACDIS program or other follow on programs. The template files are labeled %up%udeg. and the test files are labeled %up%utst. Where the first %u stands for the whole portion of the AOA, the p stands for the decimal point, and the second %u stands for the fractional portion of the AOA.

The output file is labeled as a template or test file depending on whether or not deg or tst files have been submitted. An example of a printable template run output file would be: template%u.doc. An example of a printable test run output file would be: test%u.doc. The %u in either of the output files stands for an operator set run number.

D.1 Running the Program

The XFM program is started by typing xfm. The program is driven by questions asked of the operator.

The program first asks for a run number. This number is an arbitrary decimal number created by the operator to keep track of different sets of data files.

Next the program asks "Are these files template ... y/n?" If the answer is yes then the program will look for %up%udeg. files. If answered no then the program will look for %up%utst. files. The yes also sets the output file as a template file. The no sets the output file as a test file.

Third the program wants to know the first whole angle of the sequential data files to process. For example, if the operator wishes to process a set of data files representing angles between 0 and 10, the operator would enter 0 in response to this question.

Fourth the program wants to know the last whole angle of the sequential data files to process. For example, if the operator wishes to process a set of data files representing the angles between 0.0 and 0.80, the operator would enter 0 in response to this question.

The last question asked is "How many sub-angles per angle?" For example 1 for no sub-angles, 2 for 1/2 degree angles, 4 for 1/4 degree angles, or 5 for 1/5 degree angles.

Once the questions have been answered the program reads and processes the files sequentially. The resultant feature vectors made up of the dc component and the lower three harmonics of the Fourier transform of each input file are put in the output files.

D.2 Example Run

In this example the operator wishes to process the following test files: 26p0tst., 26p25tst., 26p50tst., 26p75tst., 27p0tst., 27p25tst., 27p50tst., 27p75tst. Further the operator wishes to assign the run number 20 to the resultant test files. The sequence of questions and answers to the XFM program would be:

- "Enter run number" 20
- "Are these files template ... y/n?" y
- "Whole angle of first test file?" 26
- "Whole angle of the last file?" 27
- "How many sub-angles per angle?" 4

The XFM program would print out to the screen that the resultant test vectors will be stored in files test20.vec and test20.doc. The resultant test vectors can be viewed by printing or sending to the screen the test20.doc file. The program also prints to the screen the test or template file name as the program processes the individual file.

```
/* XFM.C
```

This program is to be used with Spiricon files (i.e. 1deg. or 1p5deg.)

INPUT - %udeg.256 or %utst.256 where %u is the angle
- number of sub-angles per deg

OUTPUT template%u.vec/.doc or test%u.vec/.doc where %u
is the run number.

```
*/
```

```
#include math  
#include string  
#include stdio
```

```
#define size 512  
#define nsize 256  
#define hsize 128  
#define bsize 124  
#define esize 132  
#define vsize 125
```

```
char filename_in[80], filename_vec[80], filename_doc[80];  
char template;  
unsigned outer, inner;  
unsigned char data_in[size][size];  
float array[nsize*nsize*2], real, imag, magn[nsize][nsize];  
float matrix_out[nsize][nsize], mag;  
float array_in[nsize][nsize], vector[7][7];  
int i, j, nn[3], angle_b, angle_e, num_files, cnt, run, cnt_b;  
int itch, itcha;  
unsigned run_num;
```

```
main()  
{  
FILE *infile, *outfile, *docfile;
```

```
/* SET UP FOR NUMBER OF FILES TO READ IN */
```

```
printf("\n\nXFM.C ");
```

```

printf("\n\nEnter run number > ");
scanf(" %d", &run);
run_num = (unsigned)run;

printf("\n\nAre these files template ... y/n? ");
scanf(" %c", &template);
if (template=='y') {
    printf("\n\nWhole angle of first template file? > ");
}
else {
    printf("\n\nWhole angle of first test file? > ");
}
scanf(" %d", &angle_b);
printf("\n\nWhole angle of last file? > ");
scanf(" %d", &angle_e);

printf("\n\nHow many sub-angles per angle? > ");
scanf(" %d", &itcha);
itch = 100/itcha;

num_files = itcha*(angle_e - angle_b +1);

printf("\n\nNumber of files to process = %d ", num_files);

if(template=='y') {
    sprintf(filename_vec, "template%u.vec", run_num);
    sprintf(filename_doc, "template%u.doc", run_num);
    printf("\n\nTemplate vectors will be stored in ... %s",
           filename_vec);
    printf("\n\nTemplate vectors will also be stored in ... %s",
           filename_doc);
}
else {
    sprintf(filename_vec, "test%u.vec", run_num);
    sprintf(filename_doc, "test%u.doc", run_num);
    printf("\n\nTest vectors will be stored in ... %s",
           filename_vec);
    printf("\n\nTest vectors will also be stored in ... %s",
           filename_doc);
}
/* OPEN OUTPUT FILE FOR RESULTANT VECTORS */

```

```

/*Open output .vec file */
if((outfile = fopen(filename_vec, "w")) ==NULL) {
printf("\n\nError in opening output vector file %s ",
      filename_vec);
exit(1);
}
/* Open output .doc file */
if((docfile = fopen(filename_doc, "w")) ==NULL) {
printf("\n\nError in opening output vector file %s ",
      filename_doc);
exit(1);
}
/* Print .doc title */

fprintf(docfile, "\n      filename %s", filename_doc);

/* ***** */

for (cnt=angle_b; cnt<angle_e+1; cnt++) {
    for(cnt_b=0; cnt_b<itcha; cnt_b++) {

/* SET UP INPUT FILE NAME */

outer = (unsigned)cnt;
inner = ((unsigned)(cnt_b*itch));

if(template=='y') {
    sprintf(filename_in, "%up%udeg.", outer, inner);
}
else {
    sprintf(filename_in, "%up%utst.", outer, inner);
}
printf("\nProcessing file %s ", filename_in);

/* READ IN DATA */

if((infile = fopen(filename_in, "rb")) ==NULL) {
printf("\nError in opening input file %s ", filename_in);
exit(1);
}
fseek(infile, 2, 0); /* skip over spricon header */

```

```

/* READ IN 512x512 BINARY FILE */
fread(data_in, 1, size*size, infile);

/* PULL OUT CENTER OF 512X512 FILE and FLOAT IT */

for (i=0; i<size; i++) {
    for(j=0; j<size; j++) {
        if (i>127 && i<384 && j>127 && j<384) {
            array_in[i-128][j-128] = (float)data_in[i][j];
        }
    }
}
fclose(infile);

/* printf("\nFinished reading and floating the data"); */

/* Fill Array */

for(i=0; i<nsize; i++) {
    for( j=0; j<nsize*2; j++, j++) {
        array[i*2*nsize+j+1] = array_in[i][j/2];
        array[i*2*nsize+j+2] = 0.0; /* add imag array */
    }
}

/* printf("\nFinished adding imag array"); */

/* FOURIER TRANSFORM ARRAY */

nn[1] = nn[2] = nsize;
fourn(array,nn,2,1);
/* printf("\nFinished fourier transform"); */

/* COMPUTE MAGNITUDE OF THE FFT */

for(i=0; i<nsize; i++) {
    for(j=0; j<nsize*2; j++,j++) {
        real = array[i*2*nsize+j+1];
        imag = array[i*2*nsize+j+2];
        mag = sqrt((real*real + imag*imag));
        magn[i][j/2] = mag/(nsize*nsize);
    }
}

/* FLIP MATRIX SO THAT THE DC COMPONENT IS AT THE CENTER OF THE
MATRIX */

```

```

/*      1 | 2
      -----
      4 | 3
*/
/* FLIP 1 TO 3 */
for(i=0; i<hsize; i++) {
    for(j=0; j<hsize; j++) {
        matrix_out[i+hsize][j+hsize] = magn[i][j];
    }
}
/* FLIP 2 TO 4 */
for(i=0; i<hsize; i++) {
    for(j=hsize; j<nsize; j++) {
        matrix_out[i+hsize][j-hsize] = magn[i][j];
    }
}
/* FLIP 3 TO 1 */
for(i=0; i<hsize; i++) {
    for(j=0; j<hsize; j++) {
        matrix_out[i][j] = magn[i+hsize][j+hsize];
    }
}
/* FLIP 4 TO 2 */
for(i=0; i<hsize; i++) {
    for(j=hsize; j<nsize; j++) {
        matrix_out[i][j] = magn[i+hsize][j-hsize];
    }
}
/* PULL OUT THE CENTER OF THE 2-DFFT */

for(i=0; i<nsize; i++) {
    for(j=0; j<nsize; j++) {
        if(i>bsize && i<esize && j>bsize && j<esize) {
            vector[i-vsize][j-vsize]=matrix_out[i][j];
        }
    }
}
/* SAVE THE VECTOR */

/* Save to filename_doc */

fprintf(docfile, "\n\n      Vector Number %u.%u\n", outer,
        inner);

for (i=0; i<7; i++) { fprintf(docfile, "\n      ");
    for (j=0; j<7; j++) {
        fprintf(docfile, " %f", vector[i][j]);
    }
}

```

```
}}  
fwrite(vector, 4, 49, outfile); /* Save to filename_out */  
}  
}  
/* ***** */  
fclose (outfile); /* close the vector file */  
fclose (docfile);  
  
}
```

Appendix E. Subroutine FOURN

This appendix provides the source code for FOURN subroutine. The subroutine FOURN, when called by another routine, takes a real, 2^n , multi-dimensional array and performs a Fourier transform on the array. The input array must contain zeros in the imaginary elements (refer to Figure 23). Figure 24 shows the structure of the Fourier transform output array. The Fourier transform output is saved back to the same filename. In other words, the output is written over the input.


Input	Array	Sample Time
(1)	real	t=0
(2)	imag	
(3)	real	t=1
(4)	imag	
		
(2N-3)	real	t=(N-2)
(2N-2)	imag	
(2N-1)	real	t=(N-1)
(2N)	imag	

Figure 23. Input Array (13:411)

The original source of this subroutine is from the book *NUMERICAL RECIPES in C The Art of Scientific Computing* (13:407-412). The version given here was modified for n-dimensional arrays by Captain Gregory Tarr.

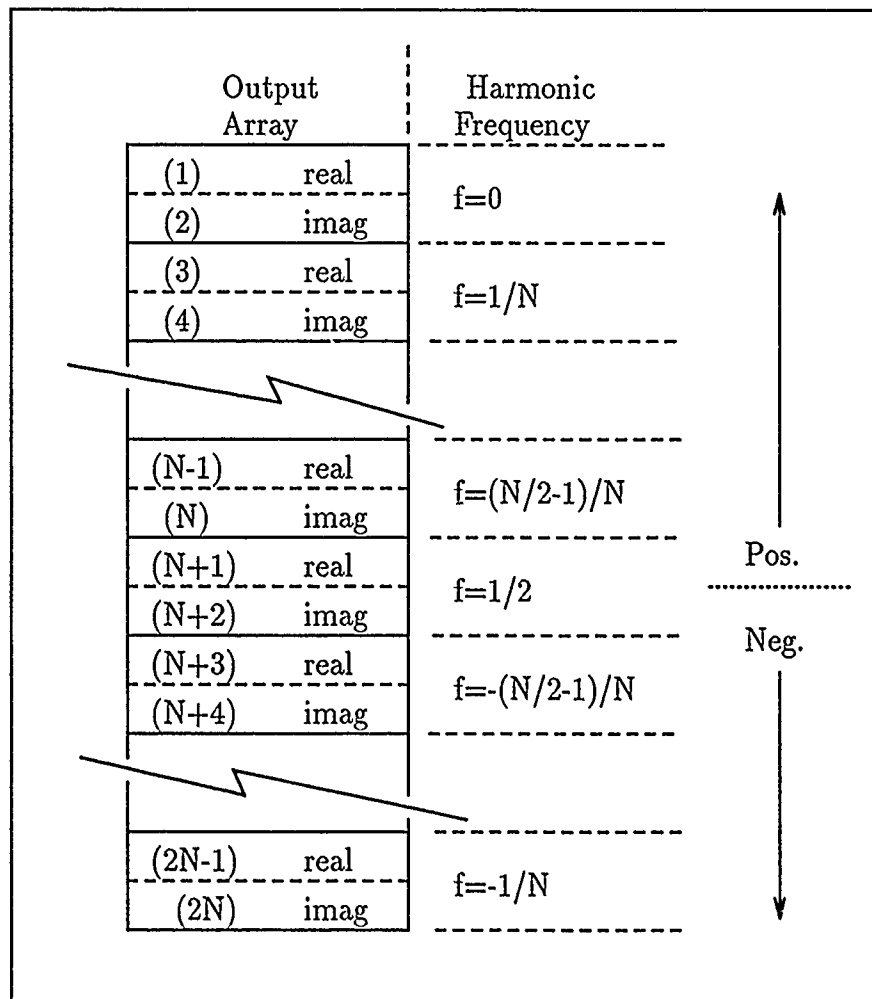


Figure 24. Output Array (13:411)

FOURN.C

/* Replaces data by its complex dft

From the "NUMERICAL RECIPES in C The Art of Scientific Computing" ISBN 0-521-35465-X

Modified for n-dimensional arrays by
Captain Gregory Tarr

inputs

data - address to array
nn - nn[3] = {0, nsize, nsize}
ndim - dimension of Fourier transform required
isign - +1 or -1
+1 = FFT
-1 = inverse FFT (must multiply result
by 1/N) ref eqn 12.1.9

data format

in		out
1 real	to	1 real fo (dc term)
2 imag	to	2 imag fo
.		.
.		.
.		.
2N-1 real	to	2N-1 real
2N imag	to	2N imag

*/

#include <math.h>

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

void fourn(data,nn,ndim,isign)

float data[];

int nn[],ndim,isign;

{

int i1,i2,i3,i2rev,i3rev,ip1,ip2,ip3,ifp1,ifp2;

int ibit,idim,k1,k2,n,nprev,nrem,ntot;

float tempi,tempr;

double theta,wi,wpi,wpr,wr,wtemp;

ntot=1;

for (idim=1;idim<=ndim;idim++)

ntot *= nn[idim];

```

nprev=1;
for (idim=ndim;idim>=1;idim--) {
    n=nn[idim];
    nrem=ntot/(n*nprev);
    ip1=nprev << 1;
    ip2=ip1*n;
    ip3=ip2*nrem;
    i2rev=1;
    for (i2=1;i2<=ip2;i2+=ip1)\{
        if (i2 < i2rev) {
            for (i1=i2;i1<=i2+ip1-2;i1+=2) {
                for (i3=i1;i3<=ip3;i3+=ip2) {
                    i3rev=i2rev+i3-i2;
                    SWAP(data[i3],data[i3rev]);
                    SWAP(data[i3+1],data[i3rev+1]);
                }
            }
            ibit=ip2 >> 1;
            while (ibit >= ip1 \&\& i2rev > ibit) \{
                i2rev -= ibit;
                ibit >>= 1;
            }
            i2rev += ibit;
        }
    }
    ifp1=ip1;
    while (ifp1 < ip2) {
        ifp2=ifp1 << 1;
        theta=isign*6.28318530717959/(ifp2/ip1);
        wtemp=sin(0.5*theta);
        wpr = -2.0*wtemp*wtemp;
        wpi=sin(theta);
        wr=1.0;
        wi=0.0;
        for (i3=1;i3<=ifp1;i3+=ip1) {
            for (i1=i3;i1<=i3+ip1-2;i1+=2) {
                for (i2=i1;i2<=ip3;i2+=ifp2) {
                    k1=i2;
                    k2=k1+ifp1;
                    tempr=wr*data[k2]-wi*data[k2+1];
                    tempi=wr*data[k2+1]+wi*data[k2];
                    data[k2]=data[k1]-tempr;
                    data[k2+1]=data[k1+1]-tempi;
                    data[k1] += tempr;
                }
            }
        }
    }
}

```

```

                                data[k1+1] += tempi;
                                }}
                                wr=(wtemp=wr)*wpr-wi*wpi+wr;
                                wi=wi*wpr+wtemp*wpi+wi; }
                                ifp1=ifp2; }
                                nprev *= n;
                                }}
                                #undef SWAP

```

Appendix F. Program NACDIS

This appendix provides the 'C' source code for the program NACDIS. The NACDIS program reads in template and test files generated by the XFM program and performs a 24 component Euclidean distance measurement on each test feature vector verses each template feature vector. The distance measurements are stored in the file labeled %u_nacdis_%u.doc. The first %u corresponds to the run number of the template%u.vec file, and the second %u corresponds to the run number of the test%u.vec file.

Only 24 of the 49 component feature vector from the XFM program is used by this program. Since the input data is symmetrical about the dc component of the feature vector the last 24 components are ignored so as to speed processing time. The dc component is ignored since leaving it out improved the distance measurement results. The NACDIS program normalizes each 24 component feature vector before computing the Euclidean distance between vectors.

F.1 Running the Program

The program is started by typing nacdis. The program is driven by questions asked of the operator.

The first the program asks for the run number of the template file. Next, the program asks for the run number of the test file.

Both inputs must be decimal numbers corresponding to the run numbers of the two files. Third the program requests the first and last whole angles of the original deg files used to produce the template file. The fourth request is for the first and last whole angles of the original tst files used to produce the test file. The last request is for the number of sub-angles per degree.

The number of sub-angles per degree and the number of whole angles allows the program to compute how many test files it will have to process. The program prints to the screen how many test file it has to process, and the file to which the output will be written to. The program then computes the distances between each of the template vectors and each of the angles. The resultant lowest distance between each test and template vector is printed to the screen. All distances between each of the template vectors are stored in the output file. The output file is in ASCII format so it can be printed or view directly on the screen.

F.2 Example Run

In this example the operator wishes to process the following template and test files: template20.vec and test20.vec. For this example the template20.vec file was created by the XFM program for the following template files: 0p0deg., 0p50deg., 1p0deg., and 1p5deg. And the test20.vec file was created by the XFM program for the following test files: 0p0tst., 0p50tst., 1p0tst., and 1p50tst. The sequence of questions and answers to the NACDIS program would be:

- "What is the run number of the template file?" 20
- "What is the run number of the test file?" 20
- "The results will be stored as 20_nacdis_20.doc".
- "What is the 1st whole template AOA?" 0
- "What is the Last whole AOA?" 1
- "What is the 1st whole test AOA?" 0
- "What is the Last whole AOA?" 1
- "How many sub-angles per AOA?" 2
- The program will print out "Number of test files to process = 4".

- The program will process the data and print out "Vector %u indicates AOA of %u deg" four times (once for each test vector).

The first %u of the above statement is the test vector number, and the second %u is the angle of the closest template vector. If the above template and test vectors matched their respective angles with the smallest Euclidean distance measurement then the output would be as follows:

- "Vector 0. 0 indicates an AOA of 0. 0 deg"
- "Vector 0.50 indicates an AOA of 0.50 deg"
- "Vector 1. 0 indicates an AOA of 1. 0 deg"
- "Vector 1.50 indicates an AOA of 1.50 deg"

A complete listing of distances would be given in the 20_nacdis_20.doc file along with the closest match (smallest distance).

```
/* NACDIS.C
```

```
    This program compares the test to the template  
    vectors of the XFM program.
```

```
Input vectors are normalized before being compared.
```

```
The distance is computed on only 24 components ... no dc.
```

```
INPUTS  The program wants to know what is the run numbers of the  
        test and template vectors you wish to compare.
```

```
Input files are template%u.vec and test%u.vec where the  
%u is the file run number.
```

```
Output file is %u_nacdis_%u.doc where the first %u is the  
template run number and the second %u is the test  
run number.
```

```
*/
```

```
#include <stdio>
```

```
#include <string>
```

```
#include <math>
```

```
char file_template[80], file_test[80], file_doc[80];  
int num_test, num_template, test_b, test_e, template_b, template_e;  
int i, template_run, test_run;  
int tst_cnt, temp_cnt; tst_cnt_frac, temp_cnt_frac;  
int num_frac_degs, frac_tst, frac_temp, min_whole_vec,  
min_frac_vec;  
float test_vector[49], template_vector[49];  
float norm_test_vec[49], norm_template_vec[49];  
double sum, min_distance, distance, mag_sum, mag_sum_temp;  
double mag_vec, mag_vec_temp;  
unsigned temp_run, tst_run;
```

```
main()
```

```
{
```

```
FILE *intemplate, *intest, *outdoc;
```

```
/* READ IN NUMBER OF FILES TO PROCESS */
```

```
printf("\nNACDIS.C ");
```

```
printf("\n\nWhat is the run number of the template file? > ");
```

```
scanf(" %d", &template_run);
```



```

temp_run = (unsigned)template_run;
sprintf(file_template, "template%u.vec", temp_run);

printf("\n\nWhat is the run number of the test file? > ");
scanf(" %d", &test_run);
tst_run = (unsigned)test_run;
sprintf(file_test, "test%u.vec", tst_run);

sprintf(file_doc, "%u_nacdis_%u.doc", temp_run, tst_run);
printf("\n\nThe results will be stored as %s ", file_doc);

printf("\n\nWhat is the 1st whole template AOA? > ");
scanf(" %d", &template_b);

printf("\n\nWhat is the Last whole AOA? > ");
scanf(" %d", &template_e);

printf("\n\nWhat is the 1st whole test AOA? > ");
scanf(" %d", &test_b);

printf("\n\nWhat is the Last whole test AOA? > ");
scanf(" %d", &test_e);

printf("\n\nHow many sub-angles per AOA? > ");
scanf(" %d", &num_frac_degs);
num_template = num_frac_degs*(template_e-template_b);
num_test = num_frac_degs*(test_e - test_b + 1);
printf("\n\nNumber of test files to process = %d ", num_test);

/* OPEN TEMPLATE, TEST, AND DOC FILES */

if((intemplate = fopen(file_template, "r")) ==NULL) {
    printf("\nError in opening input file %s ", file_template);
    exit(1);
}
if((intest = fopen(file_test, "r")) ==NULL) {
    printf("\nError in opening input file %s ", file_test);
    exit(1);
}
if((outdoc = fopen(file_doc, "w")) ==NULL) {
    printf("\nError in opening output file %s ", file_doc);
    exit(1);
}

```


Appendix G. Program MZNET

This appendix provides the source code for the MZNET program. The MZNET program takes in a ten class binary output file from the XFM program and produces a class labeled ASCII file out. The ASCII output file is readable by the Radial Basis Function (RBF) neural network software written by Captain Zahirniak. The program's output is also printable, to the screen or to a printer, because the ASCII output is limited to eight characters across before a hard carriage return is added to the output file.

G.1 Running the Program

The MZNET program is started by typing `mznet`. The program is driven by questions asked of the operator.

First the program asks for a file to store the resulting ASCII file to. The program will automatically add the suffix `.doc` to the entered output file name. Next the program asks for the name of the binary input file. The program will automatically add the `.vec` suffix to the entered input file name.

The program will open a new file if the file given does not exist or will append to a previously opened file. The appending to previous files allows multiple runs to be combined into a single file to be read by the RBF neural network software. For example, all of the 50 series runs were combined into a file called `run50.doc` after running the MZNET program four times for runs: 50, 51, 52, and 53.

```
/* MZNET.C
```

```
    This program converts the output of the xfm program  
    (.vec file) into a format readable by the Zahirniak RBF  
    neural network program. The output file is also printable.
```

```
    The program requests the input and output file names. The  
    program opens a new output file or appends to a previously  
    opened file.
```

```
*/
```

```
#include math  
#include stdio  
#include string
```

```
char filename_in[80], filename_out[80];  
float array[3][8];  
int i, j, jj;
```

```
main()  
{
```

```
FILE *infile, *outfile;
```

```
/* Request file to output to */
```

```
printf("\nEnter file to output processed file to > ");  
scanf("%s", filename_out);  
strcat(filename_out, ".doc");
```

```
if ((outfile = fopen(filename_out, "a")) == NULL) {  
printf("\n problem writing to %s", filename_out);  
exit (1);  
}
```

```
printf("\nMNetZ file will be stored on disk as %s ", filename_out);
```

```
/* Request file to process */
```

```
printf("\nEnter file name to process to Znet > ");  
scanf("%s", filename_in);  
strcat(filename_in, ".vec");
```

```
printf("\nInput file name is %s ", filename_in);
```

```

if ((infile = fopen(filename_in, "rb")) == NULL) {
printf("\n problem reading %s", filename_in);
exit (1);
}
for(i=1; i<11; i++) {
    fread(array, 4, 49, infile);
    for(j=0; j<3; j++) {
        for(jj=0; jj<8; jj++) {
            fprintf(outfile, " %f", array[j][jj]);
        }
        fprintf(outfile, "\n");
    }
    fprintf(outfile, " %u\n ", i);
}
fclose (infile);
fclose (outfile);
}

```

Bibliography

1. Holl, Herbert B. *Angle of Arrival Meter*, 1982. Patten Application: S.N. 349128 (AD-D009335/9).
2. Horridge, Adrian G. "The Compound Eye of Insects," *Scientific American*, 237(1):108-120 (July 1977).
3. Interactive Intelligent Imagery Corporation. *Optical Sensor With High Directional Resolution* (SBIR Phase I Report, Topic AFF88-184 Edition). Technical Report. Menlo Park CA, March 1989.
4. Kabrisky, Matthew E. Class handout distributed in EENG 620, Pattern Recognition I. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, April 1990.
5. Kaiser, Capt Robert D. *Analysis and Test of a Wide Angle Spectrometer*. MS thesis, AFIT/GEP/ENP/89D-6, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989. (AD-A215819).
6. Keiser, Gerd. *Optical Fiber Communications*. New York: McGraw-Hill, Inc., 1983.
7. Lippmann, Richard P. "An Introduction to Computing with Neural Nets," *IEEE ASSP MAGAZINE*, 4:2-22 (April 1987).
8. Moody, John and Christian J. Darken. *Fast Learning in Networks of Locally-Tuned Processing Units*. Technical Report YALEU/DCS/RR-664, New Haven, CT: Yale Computer Science Department, October 1989.
9. Newport Corporation. *PMC300 Operator's Manual*. Technical Report. P.O. box 8020, Fountain Valley CA, 1989.
10. Newport Corporation. *Laser Collimators Instruction Manual T28 and T27 Series*. Technical Report. P.O. box 8020, Fountain Valley CA, Not Dated.
11. Nowlan, Steven J. *Max Likelihood Competition in RBF Networks*. Technical Report CRG-TR-90-2, Toronto Canada: Department of Computer Science, University of Toronto, February 1990.
12. Parker, Jack H. Jr. *Common Opto-Electronic Laser Detection System (COLDS)* (AFWAL-TR-86-1048 Edition). Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB OH, December 1986. (AD-B113557).
13. Press, William H., et al. *NUMERICAL RECIPES in C: The Art of Scientific Computing*. New York: Cambridge University Press, 1988.
14. Renals, Steve and Richard Rohwer. "Phoneme Classification Experiments Using Radial Basis Functions," *In Proceedings of the International Joint Conference on Neural Networks*, 1:461-467 (1989).

15. Welker, Capt John Wallace. *Angle of Arrival Detection Through Analysis of Optical Fiber Intensifying Patterns*. MS thesis AFIT/GE/ENG/89D, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989. (AD-A215421).
16. Zahirniak, Capt Daniel R. *Characterization of Radar Signals Using Neural Networks*. MS thesis AFIT/GE/ENG/90D, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1990. (DTIC number not available at this time.).

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1990	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Angle of Arrival Detection through RBF Artificial Neural Network Analysis of Optical Fiber Intensity Patterns		5. FUNDING NUMBERS	
6. AUTHOR(S) Scott Thomas, Captain, USAF			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/90D-62	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This thesis demonstrated that an intensity pattern out of a short piece of optical fiber could be used to determine the angle of arrival (AOA), to within 0.1deg, of the incident laser energy on the front of the optical fiber. The optical fiber was a one-inch-long, 3mm-diameter, multimode, step-index, plastic fiber. The optical fiber was mounted to the front end of a charge injection device (CID) camera. The CID camera's angle with respect to the incident laser energy, a uniform amplitude plan wave, could be varied by a computer controlled rotational stage. The output of the CID camera was captured by Spiricon software. Captured outputs representing various AOAs were processed to provide template or test feature vectors. The processing method used a fast Fourier transform routine to create a 24 component low frequency feature vector. Two classification methodologies were used, a Euclidean distance method and a radial basis function (RBF) neural network.			
14. SUBJECT TERMS Angle of Arrival, Optical Fibers, Pattern Recognition, Radial Basis Function, Artificial Neural Networks		15. NUMBER OF PAGES 104	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL